

**FREE
48-PAGE
SUPPLEMENT**

Volume 6 Issue 10



Denmark DKr34.50, Greece Dra245.00, Holland DFL8.50, Italy L4100, Spain Pts360.00, Switzerland SFr7.40, Germany DM8.50, France Fr32.60, Canada C\$4.50, Australia A\$3.00, Singapore M\$6.35

LIST

PRACTICAL COMPUTING OCTOBER 1983

>NEWS

15 HARDWARE NEWS

Spectrum microdrives arrive at last. Newbrain in trouble, plus Apple plotter.

21 SOFTWARE NEWS

Apple launches a new DOS, and Olivetti comes in from the cold.

61 IBM PC NEWS

PC dongles from Wordcraft, and a glut of RAM expansion cards.

>REVIEWS

60 IBM PC XT DESK-TOP TEST

Part 3: the features of DOS 2 and running software on the hard disc.

64 16-BIT REVIEW LOGICA VTS

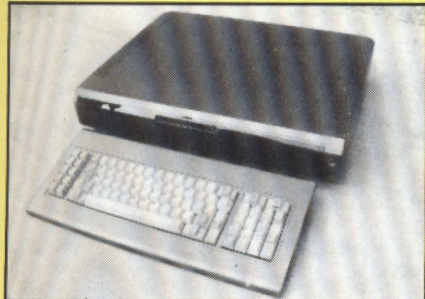
Chris Bidmead checks out a British-made micro designed for office use.

68 ACORN ELECTRON — A BBC KILLER?

Is Acorn's £199 offering compatible enough, and versatile enough, to replace the BBC Micro? Neville Maude reports.

72 THE ADVANCE — AN IBM PC KILLER?

Ian Stobie previews a new British machine promising IBM PC compatibility starting from around £350.



78 CP/M PLUS

John and Timothy Lee investigate the new Digital Research replacement for CP/M 2.2.

82 DRAGON GAMES

We tried two dozen awful games for this popular micro. Ian Stobie reports on the best of a bad lot.

86 HOME WP —PART 7 ATARIWRITER

Word processing on ROM for the Atari 400 and 800, tested by Jack Schofield.

167 IBM PC BOOKS

Books for the PC are coming out fast. Jack Schofield reviews a baker's dozen.

>VDU'S AND GRAPHICS

93 SETTING THE SCREEN

An introduction to this month's special topic, computer graphics and things to display them on.

94 VDUs, MONITORS AND TV SCREENS

Chris Naylor describes how visual display units work, and the advantages and disadvantages of TVs and monitors.

102 YOUR GUIDE TO GRAPHICS

John Lewell looks at recent developments in computer graphics and provides an overview of the market.

108 GEOMETRICAL PLOTING LIBRARY

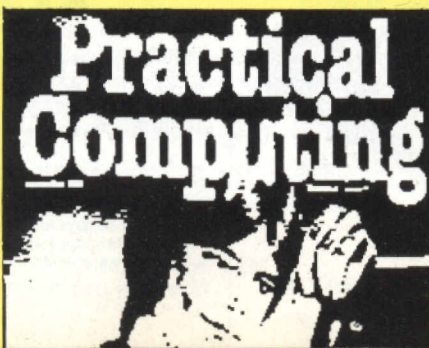
Jonathan Bowen's routines can be used for graphics drawing in almost all versions of Basic.

112 AREA FILLING ON THE BBC MICRO

John G Dallman reveals some sophisticated area-fill techniques for the BBC computer.

116 DIGITISING CAMERA INPUT

Peter Kruger and Stephen Cronk of Dighurst describe how pictures can be converted into screen graphics using low-cost image-analysis techniques.



>FEATURES

86 PC's BIG GAME HUNT

Rate your favourite game and send it in for our Special Games Issue in December.

118 FICTION — DEATH TO THE MACHINE

122 STATISTICS — COFFEE, TEA OR?

Survey a range of choices: when do the results become meaningful? Owen Bishop explains.

126 EDUCATION — CASH REGISTER

Commerce comes to the classroom in Hewan Ormson's interesting programs.

130 APPLICATIONS — POST CODES

Post codes can be a profitable key for mailing list/address sorting.

>REGULARS

5 EDITORIAL — ONE MORE MICRO MAGAZINE

The launch of *Computer Choice*, and thoughts on the choice of magazines.

7 FEEDBACK YOUR LETTERS

Correspondence, corrections, love letters and advice.

29 CHIP-CHAT

Ray Coles on networks which use public telephones.

35 RANDOM ACCESS METAMATHEMATICS

Boris Allan continues his discussion of threaded interpretive languages.

141 OPEN FILE PROGRAMS

Sixteen pages of free software for Apple, BBC, Sinclair, and other popular micros.

179 LAST WORD OBFUSCATION

Why use one word when ten will do? Chris Naylor tells you how to master TechnoSpeak.

This month's cover illustration was created by Steve Miller and Ian Stobie using a Hewlett Packard Series 200 Model 16 with HP-7470A plotter.

Editor
Jack Schofield
Deputy Editor
Bill Bennett
Assistant Editor
Ian Stobie
Art Editor
Steve Miller
Production Editor
John Liebmann

Sub-editor
Sally Clark
Editorial Secretary
Sue Jordan
Consultants
Chris Bidmead
Peter Laurie

Advertisement Manager
Ian Carter 01-661 3021
Assistant Advertisement
Manager
Kenneth Walford 01-661 3139
Advertisement Executives
Lynne Brennan 01-661 3468
Robert Payne 01-661 8425
David Honeyman 01-661 8626

Advertisement Secretary
Janet Thorpe
Midlands office:
David Harvett 021-356 4838
Northern office:
Geoff Aikin 061-872 8861
PUBLISHING DIRECTOR
Chris Hipwell

Published by Electrical Electronic
Press, Quadrant House, The Quadrant,
Sutton, Surrey SM25AS. Tel: 01-661
3500. Telex/grams 892084 BISPRS G.

Distributed by Business Press
International Ltd, Quadrant House,
The Quadrant, Sutton, Surrey SM2
5AS

Subscriptions: U.K. £13 per annum; Overseas £19 per annum; selling price in £ire subject to currency exchange fluctuations and VAT; airmail rates available on application to Subscription Manager. Business Press International Ltd, Oakfield House, Perrymount Road, Haywards Heath, Sussex RH16 3DH. Tel: 0444 459168.

Printed in Great Britain for the
proprietors Business Press
International Ltd by Eden Fisher
(Southend) Ltd, Southend-on-Sea.
Typeset by Centrepont Typesetters,
London EC1

© Business Press International Ltd
1983

Would-be authors are welcome to send articles to the Editor but *PC* cannot undertake to return them. Payment is at £35 per published page. Submissions should be typed or computer-printed and should include a tape or disc of any program. Hand-written material is liable to delay and error.

Every effort is made to check articles and listings but *PC* cannot guarantee that programs will run and can accept no responsibility for any errors.

Overkill

Of course both moans are sweeping generalisations, and all generalisations are false. There may be too many badly designed micros, but there is still room for a few more good ones. Similarly there may be too many magazines, but there is surely a market for one or two more with something new to offer.

The last bookstall magazine launched from the *Practical Computing* office was *Your Computer*. Its brief was to concentrate on the home/games market, leaving us free to focus on the more serious side of computing. With the boom in the home market *Your Computer* has grown to be the U.K.'s largest selling micro magazine — by a wide margin.

Naturally we are delighted with this response to our efforts, if only because it proves there are people out there who are interested in more than just space invaders. Needless to say we will try to make the magazine even more useful and informative in the future.

Normally we do not boast about our small successes, but the magazine market is becoming very competitive. With new micros being launched there are always new potential readers and new potential advertisers. Almost by definition they start from a position of ignorance.

The things they say make us laugh. If you owned one of their micros or were employed making them they might make you cry. The reason is that in the next few years some 300 of the 400 companies making and/or distributing micros in the U.K. are likely to be taken over or go out of business. Designers and product managers will wonder where they went wrong, after all, their micro was as good as, or better than, the next one — right? They spent enough money on promotion — right? So why didn't they sell? Well we know the answer already, but in this case there is little joy in being wise before the event.

The forthcoming shake-out of manufacturers will lead to a shake-out of magazines, partly because not all of that misplaced advertising will be paid for. Also, the more aware companies will start to gauge the response they get to their advertisements, and — at last — the number of leads that are converted into sales. The many worthwhile magazines, including *Practical Computing*, *Your Computer* and, we trust, *Computer Choice* will continue to prosper while the rest quietly sink away.

All this will make the world a more reliable but, some would say, a duller place. We do not agree. Microcomputing is never going to be a dull subject. M

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

5 Years ago ...

[illegible]

A hobby computer just coming to the market in the U.K. is the Sorcerer. It is made by the American firm, Exidy Inc. a video games manufacturer.

own power supply, TV monitor and cassette tape recorder(s).

The main selling point of the Sorcerer is price. You can buy a 16K version for £760 or a 32K version for £950 and, considering its facilities, this represents good value for money.

For expansion purposes it takes the S-100 bus which gives you the ability to interconnect large memories, disc drives, speech and communications facilities.

The system loaned to us was the 32K version. Standard configuration includes a 61-key typewriter keyboard and a 16-key pad. It looks like the Tandy keyboard without the numeric pad. To that you add your

The striking feature of the Sorcerer, though, is the way you load the Basic. It's a standard Basic which is loaded by way of a cartridge into the side of the keyboard. It looks rather like an eight-track stereo cartridge but inside is a ROM containing the language.

Practical Computing Volume 1, Issue 4

[illegible]

Pascal semicolon

IN YOUR JUNE 1983 issue, page 7, John Robinson writes that the statement

```
if x = y then if w = z then a = 1;
    else b = 1;
```

is a correct Pascal statement. This is not true, see the book by Jensen-Wirth, *Pascal. User Manual and Report*, page 26, "Caution: there is never a semicolon before and else." Hence, the text:

```
if p then begin S1; S2; S3 end; else S4
```

is incorrect. Perhaps even more deceptive is the text:

```
if p then; begin S1; S2; S3 end
```

Here, the statement controlled by the if is the empty statement between the then and the semicolon; hence, the compound statement following the if statement will always be executed.

The syntactic ambiguity arising from the construct:

```
if <expression-1> then if <expression-2> then <statement-1>
    else <statement-2>
```

is resolved by interpreting the construct as equivalent to

```
if <expression-1> then
    begin if <expression-2> then <statement-1>
        else <statement-2>
    end
```

Hence, the correct form of the statement above is:

```
if x = y then begin if w = z then a = 1 end
    else b = 1
```

or

```
if x = y then begin if w = z then a = 1; end
    else b = 1;
```

In the second case, there are two statements between begin and end: the statement if w = z then a = 1 and an empty statement. In both cases, there is not a semicolon after b = 1 because "Pascal uses the semicolon to separate statements, not to terminate statements; i.e. the semicolon is NOT part of statement." Jensen-Wirth, *Pascal. User Manual and Report*, page 22.

Katalin Bauer,
Budapest,
Hungary.

Formcalc

BRIAN LAW'S excellent program in the July and August issues is going to be very useful to me. However I have two difficulties.

First, in the example shown in the article when entering the formula shown under the RF command — I summed column I(Load) first — results in Error 2/1650 repeatedly. Formulae of the type $K1 * K2 * K3 / 4$ work very well, but as soon as I use powers in something like $K1 * (K2 * K3) / 4$ — again meaningless except as an example — I get Error code C/1650.

Cursor shift 8, column shift to the right, does not work but cursor shift 5, to the left, does.

Can you throw light on this

for an elementary programmer like me? Incidentally, I have altered line 2305 to give results to four decimal places — it works very well.

Leon Jeavons,
Birmingham.

Brian Law replies:

It is difficult to debug programs without having the tape itself. The most likely explanations are:

- line 1840 probably has the ** missing;
- line 1310 probably has = "B" instead of = "8"

Changing the number of

Formcalc.

```
25 LET DP=2
296 IF I*(1 TO 2) = "DP" THEN GOTO 2500
2305 PRINT AT R1+2,C(V);(INT(Q(R,C)*(10**DP)+.5)/(10**DP))
2405 PRINT AT 20,C(V);(INT(Q(N,C)*(10**DP)+.5)/(10**DP))
2500 REM CHANGE DECIMAL PLACES
2505 LET DP=VAL I*(3)
2510 GOTO 1315
```

decimal places can be done on a more permanent basis using the amendments below. To change the number of places now, enter DP3 to get three places of decimals, or DP4 to get four, or DP0 to get none, etc.

Basicode plea

I WOULD like to draw everyone's attention to a new Basic language called Basicode-2, which creates a way to exchange software between different computers. The computers are Apple II, BBC Micro, Commodore Pet 2001 and Vic-20, CP/M systems, DAI, Exidy Sorcerer, Ohio Superboard, Philips P-2000, Sharp MZ-80, SWTPC-68000, Tandy TRS-80 and Video Genie.

The Basicode-2 language contains statements which are the same for all the computers. By using a translation program, which is different for all of them, your micro can understand them. If you have a program in your own Basic you can change it to Basicode-2 by using another translation program; so by using this language you can share your neighbour's programs. I think *Practical Computing* should use Basicode-2 in Open File so more people can use the programs.

Basicode-2 has been developed by NOS Hobby-scoop. It has a program on Radio Nederland every Sunday evening from 19.15 to 19.45 on 747KHz medium wave, and each week it broadcasts a Basicode-2 program. You can order the translation programs and some Basicode-2 programs, plus a manual for 30 florins — about £6. The address is NOS-Hobbyscoop, PO Box 10, 1200 JB Hilversum, Nederland. Alternatively write to me.

Michel Smit,
Zwaagdijk 152D,
1683 NN Zwaagdijk-oost,
Nederland.

The editor replies:

The manual is in both English and Dutch. Basicode is also used by Jonathan Marks on his English-language programme, Media Network, broadcast on

Thursday nights on the Dutch International Service, and rebroadcast on the short wave world wide. So far 1200 baud has proved too much for short wave use, and experiments are continuing at 300 baud. For details contact Jonathan Marks at Radio Netherlands, PO Box 222, 1200 JG Hilversum, The Netherlands.

Incidentally, payment for Basicode must be in Dutch Guilders and payable to Nos Algemeen Secretariaat. The book and cassette weigh 370g, so send 25 florins plus appropriate postage.

Practical Computing has followed the progress of Basicode with interest, but we have had no requests for coverage from outside the Netherlands. Open File programs would be easier to translate if people wrote more structured programs with sufficient REM's to give outsiders a chance. However, machine-specific tricks seem more popular.

Calculating PI

I WRITE with reference to R A Fairthorne in the Feedback section of the August issue. He seems to have taken my criticism, which I hoped was constructive, to heart. I was merely wondering why he wishes to approximate PI using such a long and tedious division.

Perhaps there is something to be gained from using his method, an unforeseen advantage. It may be faster, depending on the computer he is using, but I think I would rather enter the value of PI directly than use his division. The advantage of using my method (4*ATN(1))

is that it is easy to remember and will evaluate to as many places as the computer can handle.

Perhaps you could have a competition to find the fastest and easiest method of calculating PI? Does anybody know what the exact value of PI is?

S Mehew,
Lanarkshire,
Scotland.

Reader survey

I AM WRITING to you for assistance in writing a series of articles to illustrate how micro (continued on next page)

(continued from previous page)
computers are playing an increasingly valuable role as a low-cost aid to management and as fast information providers.

Can I ask readers to write to me with their experiences of installing a micro at work, be it a ZX-81 or an IBM. What problems have they encountered in software and hardware? Did the salespeople know what they were selling? Did, and was the buyer aware of the limitations of the computer he was being offered? Was the software adequate for the task it was bought for? If not, what difficulties were encountered to get the software or hardware working correctly?

What questions, in hindsight, would readers ask the salesperson if they had the opportunity to purchase again? Finally, and I think most important, what benefit have they received by installing a micro?

I would be most grateful to readers if they would write to me, all replies will be acknowledged by return of post.

Tom McGowran,
Monmouthshire Beacon,
50, Monnow Street,
Monmouth NP5 3XJ.

Epson solution

IN HIS ARTICLE on the Epson FX-80, August issue, page 77, Chris Roper mentions the warning in the Epson manual about control codes that cannot be sent out by certain versions of Basic. Chris points out that Epson do not propose a solution to this problem.

The authors of the Epson manual must have had in mind such quirks as CHR\$(9), which Microsoft Basic interprets as a tab character which it expands to a string of spaces.

The solution is easy, at least it

is if you are using an RS-232 interface with mark parity. You simply set the high-order bit of the control character to one. The easiest way to do this is to add 128 to the number. Thus CHR\$(9) becomes CHR\$(137). The Basic interpreter does not recognise this as a tab; the interface strips the high-order, parity, bit; and the Epson, or other output device, receives CHR\$(9) — so everyone's happy.

Mike Lewis,
London NW3.

Keen on sprites

IN YOUR JULY issue you gave a very useful program for editing sprites on the CBM 64 called 64 Sprite Editor. Being somewhat of a novice myself I was keen to utilise this program as very few magazines seem to publish, little if anything for the 64 anyway — hint, hint.

I did everything to the letter. That is, I turned the computer off and on, entered

Poke 2560,0 Poke 44, 10 return and started typing from line 30. However, immediately on pressing return after finishing line 30, the thing just crashed and the keyboard was completely disabled.

I blamed myself for this error and tried again — and again and again, repeating the instructions as per Mr Irving's article. Still no luck, so am I to blame or is there something else Mr Irving should have mentioned? Can you help — please.

Finally, I enjoy your magazine but you seem slightly

biased towards BBC, Tandy, Apple, etc.

E G Reynolds,
Lancashire.

Blunders

IN THE ARTICLE on programming sprites on the Commodore 64, in page 99 of the July issue, we unfortunately missed out two important instructions. The two Pokes entered after turning the machine on should be followed by New <Return>.

We would also like to repeat that lines 10 to 23 must be typed exactly as listed, the important feature being the number of characters entered. In the August issue, Atari Open File, page 145, line 32115 of Les Kneeling's Slow Lister program should have ended "Poke 842,12".

Indian user club

WE HAVE formed a home computer user's club in India. We meet twice a month to exchange the latest news and to try and solve members problems. Owners/users of any home computer are welcome. We have developed a music program for the ZX-81 and a battery back up. In the near future we hope to bring out a 64-column card for the ZX-81.

Arun K Nath,
New Delhi,
India.

BBC corrections

I ENTERED the disassembler program, *Practical Computing*, January 1983, in my BBC micro with the modifications printed in the March 1983 issue. I have

two corrections to communicate to other readers.

First, line 260 seems to be a little unlucky; it may have been printed a first time without a part and then wrongly corrected by M Cresswell in the March issue. The definitive — I hope — version will be:
That is the 10th value has 4 " " because the branch instruction set is printed with a 16-bit destination address.

Secondly, the addition by E Ibbotson contains a mistake; in fact, line 1520 must end with
MO\$ = OS\$(1%)

instead of

MO\$ = OS(1%)

That is MO\$ must contain the name of the OS call, not the address.

With these corrections the disassembler runs very quickly and with a very nice editing.

P Jenne,
Milan,
Italy.

Spectrum Scrabble

IN HIS REVIEW of computer Scrabble for the Spectrum, Bill Bennett expressed doubt on the validity of four particular words.

According to Chamber's 20th Century Dictionary, the national Scrabble championship's standard reference guide, while "reiner" cannot be found, definitions of the other three read:

agaze, (arch) adj. and adr. at gaze, gazing
noon. — v.i. to rest at non. — n. noon — (esp. U.S.) a repast or rest about noon.

(continued on page 13)

BBC correction.

```
260 DATA "A", "#&...", "&...",  
" &...X", "&...Y", "&...",  
" &...X", "&...Y",  
" &...", "<&...X", "<&...Y",  
"<&..."
```



Our Feedback columns offer readers the opportunity of bringing their computing experience and problems to the attention of others, as well as to seek our advice or to make suggestions, which we are always happy to receive. Make sure you use Feedback — it is your chance to keep in touch.

(continued from page 8)

tyre — v.t. to put a tyre on. —
n. tyreing

Ian Tresman,
Elstree,
Hertfordshire.

Bad timing

IT WAS WONDERFUL to see that the Newbrain was at last given a place in your excellent magazine. I, and many other Newbrain owners in this country, sincerely hope that this will become a regular feature.

L J Fourie,
Pretoria,
South Africa.

Logo

I AM PREPARING a book on the use of Logo in the classroom. The book is primarily aimed at primary school teachers, but will have some relevance to lower secondary school as well.

I would be very interested to hear of any experiences teachers have had using Logo in the classroom, and ways and means they employ to introduce the skills and concepts of computing to different age groups of children. This can include games, etc. All contributions will be acknowledged and postage refunded.

A P Mullan,
54 Copse Road,
Plympton,
Devon.

Euromouse

I AM GLAD that my write up on the Computer Fair Euromouse heats did not miss the boat entirely, despite its long delay in the post. I am sorry that it had to be cut down to fit the remaining space; particularly sorry that an acknowledgement of the Judges' efforts did not appear.

Professor Harry Prime of Birmingham University, Chairman of the Computing

and Control Division of the IEE did a splendid job of ensuring technological fairness. While Brian Glover, well known TV actor and the voice behind the Tetley Tea folk asked the contestants some searching questions, Chris Hipwell, publisher of *Practical Computing*, lent an air of authority to the judging.

I have already had an encouraging response to the announcement of a robot ping pong contest, which appeared in May 1983 issue of *Practical Computing*. Over two dozen letters have arrived including one from South Africa and one from Nato headquarters. They stress that their robot will not be an official project, nevertheless, if this letter is read by a robot enthusiast in the Kremlin we might see a needle match.

John Billingsley,
Portsmouth Polytechnic.

Loading trick

I MUST SAY how very much I enjoyed the maze program by Andrew Armstrong in the August issue of *Practical Computing*. There is no need, however, for you to exclude it from your games-of-the-year disc. I find that a short loading program seems to do the trick without causing any problems.

Give the program a suitable name, for instance, Mazel and save it on disc along with the main maze program. To use, Chain "Mazel" which will then automatically load then reload the main program. You will get an error message "Bad Mode at line 20". Ignore this, type Run and press Enter. The program should then run perfectly.

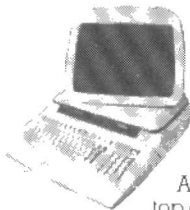
I expect other readers will have devised other methods, but I hope you find this useful.

R Dent,
Harrow,
Middlesex. ☐

Loading trick.

```
10 *KEYO LO."MAZE"!M*TAFE12!MF.T=0 TO TO
P-PAGE STEP 4:T!&EOO=T!PAGE:N.T!PAGE=&EO
O!MRUN:M
20 *FX13B,0,12B
```

You're just one step away from one-step accounting



Anagram Systems' Integrated Accounts is, simply, the most comprehensive, easiest to understand integrated accounting package available to Commodore users. It is the best reason yet for choosing Commodore.

And now there is Anagram Integrated Accounts with integrated Stock Control — on top of superb Anagram standalone Stock Control and ledger packages. Just look at these features and options.

Full integration: Anagram Integrated Accounts consists of Sales, Purchase and Nominal Ledgers, Cash Book, and functions which permit journal entries, file maintenance and report printing in remarkable detail. All these are automatically updated when you make an entry. In the version with Stock Control, creating an invoice updates the stock as well as the other modules.

Single- or multi-user: Anagram Integrated Accounts, with or without the integrated Stock Control, can be supplied either single-user, or multi-user for up to five machines for only £300 extra per machine. The same goes for Anagram Stock Control.

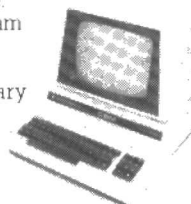
Hard disk or floppies: Integrated Accounts is on just one floppy disk — compare that to competitive packages — and runs successfully with the Commodore twin disk units. Integrated Accounts with Stock Control and Sales Order Processing ideally requires a hard disk.

Open Item or Balance Forward: Choose which system you want for each account — you can even mix the two.

Extensive analysis: Each sales invoice can be analysed across ten nominal headings; each purchase invoice across eight. You can set up budgets within each nominal heading to get comparisons of 'budget to actual' at any time.

Easy to understand, learn and use: Anagram Accounting and Stock Control packages are designed for busy people who don't want to mess about with computers. They use ordinary book-keeping terms without jargon.

Your Commodore dealer should have these packages ready for demonstration. If not, phone Anagram and we will fix a demonstration for you. Or send the coupon.



To: Actionline Sales
Anagram Systems, 60A Queen Street,
Horsham, West Sussex RH13 5AD



Send me brochures on the Anagram packages I have ticked and tell me where to see them running.

☐ Integrated Accounts with Stock Control ☐ Integrated Accounts
☐ Standalone Stock Control ☐ Standalone Ledgers

Name _____
Position _____
Company _____
Nature of business _____
Address _____
County _____ Postcode _____
Telephone No. _____
Any existing CBM computer _____

ANAGRAM SYSTEMS
60A, Queen Street, Horsham, West Sussex
RH13 5AD Tel. (0403) 59551/50854/58153

CBM is a trademark of Commodore Business Machines (UK) Ltd

AN/PCO/10

● Circle No. 105

Expandable Spectrum host to add-ons

SPECTRUM USERS are about to be inundated with a wide range of hardware add-ons, making the most popular microcomputer into one of the most expandable microcomputers. The main product is the expansion interface known as Interface I. Interface I connects the Spectrum to the new microdrives, which are mass-storage devices derived from stringy-floppies. Inside a small case is about 20 foot of very thin video tape which hurls around at very high speeds.

Microdrives can be chained together with up to eight connected to one Spectrum at any moment. Each tape can hold up to 100K of data, but

often this is nearer the 85K mark. The main drawback of the Microdrive is that it is not possible to transfer files from one drive to another without first holding the entire file in the memory of the host Spectrum. More to the point is the fact that the tapes may not last for long so users have to be very fastidious about making back-up copies.

The other facilities of Interface I are a local area network, which is invaluable for educational users and may find interesting applications elsewhere. And an RS-232 interface which enables a real printer to be hooked-up.

In effect the Spectrum is now

ready to do some serious computing, all that is needed is the software and a real keyboard. Students and some business users will be able to have the same facilities as a proper computer, but in a more primitive form and at a much lower price.

Interface II, due to appear later this year, will provide the Spectrum with a much needed joystick port just in time for the Christmas boom which seems to be a traditional feature in this untraditional market. But what will make the second Spectrum interface essential is the port for cartridge software. Expect some breathtakingly good games to appear around the same time.

Some Spectrum users may feel that a microdrive is not enough, and wish to incorporate a real floppy disc in their system rather than the ersatz microdrive. To the rescue of these users comes Technology Research Ltd, with a knight in shining armour — the disc interface card. The card supports any Shugart drive or the new and trendy 3in. discs, as worn by all the bright young micros.

The interface costs £65 and comes with a utility disc. A complete system with two drives, power supply and a case is £285, a single disc system is £195. Telephone Technology Research Ltd, 01-699 5332. 

Grundy calls in the receiver

The future of the Newbrain micro from Grundy Business Systems is in doubt and the directors are taking steps to wind up the company. After its successful launch Grundy increased Newbrain production, but sales in fact declined creating a cash-flow problem. Further problems were created by failure to meet projected dates for the introduction of system enhancements, including CP/M. The directors were unable to find further finance, though it is still possible that they will be able to

sell the business as a going concern, or alternatively sell the design of the micro. 

Intelligent cards from Zenith

ZENITH DATA SYSTEMS has introduced a 256K RAM card for use with the Z-100 series of computers, and any other S-100 bus machine. Z-100 micros have 128K as standard and can be expanded to a total of 768K.

The card is intelligent in that it can tell whether the addressing on the bus is 16- or eight-bit, it can also recognise byte parity and low-power consumption. Because it is on

an S-100 card it can be simply inserted on to the bus by the user with a minimum of fuss. The 256K dynamic RAM card is £520, for further details contact Zenith Data Systems Ltd, Bristol Road, Gloucester GL2 6EE. Telephone: (0452) 29451. 

New 96K Lynx

THE NEW 96K version of the Computers' Lynx is now available at Laskys, Spectrum and a number of independent stores. Costing £299, the new machine also includes the printer driver circuitry omitted from the original 48K machine.

Only 37.5K of the memory is accessible to Basic, with another 24K available to machine-code while the high-resolution graphics are in use. Software is available to exploit this. One nice touch is that owners of the 48K machine can upgrade by returning their Lynx to a dealers and coughing up an extra £90. The up-grade board includes an extra 4K of ROM which has printer driving routines — missing from original model — together with some extra Basic commands. 

Fortune cuts

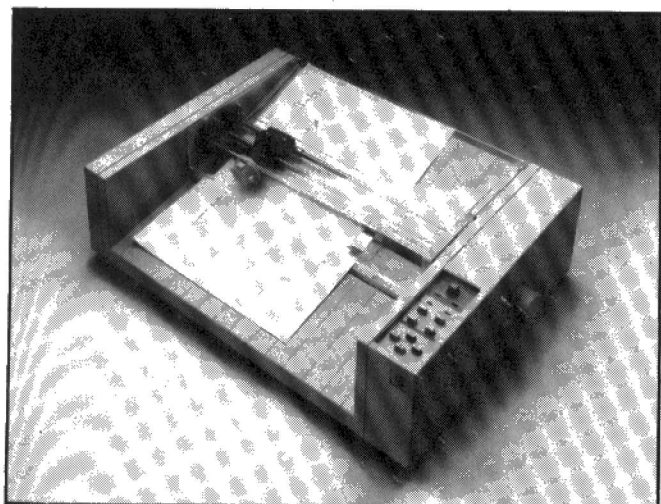
FROM THE beginning of September the price of the Fortune 32:16 computer has been cut by 35 percent. There is a wide range of models in the Fortune series including 20, 10 and 5Mbyte systems. The single-user 20Mbyte system did cost £9,245, and can now be

purchased for less than £6,000, a 5Mbyte multi-user system is reduced from £5,995 to £4,000.

IBR Microcomputers is the main Fortune distributor in the U.K. The company felt that the higher price of the Fortune meant that it had difficulty competing with other 16-bit micros in the stand-alone sector of the market. The entry-level system now costs only £3,950 and includes a 5Mbyte hard disc. It was reviewed in our January issue. 

Seven card Spectrum

THE SPECTRUM only has one expansion port, but the expansion bus system from U-Microcomputers allows up to seven cards to be used in conjunction with a special backplane. So far there are two cards that fit into the backplane, a I/O card and a serial interface, but more will follow. For more information telephone: (0925) 54117/8. 

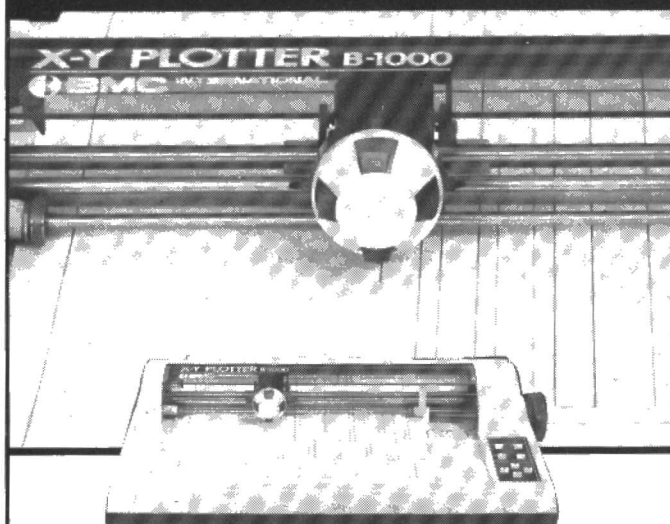


The Apple plotter has four colours and can be used with paper or transparent film in any size up to A3. Capable of working with the Apple IIe and the III the unit will set the user back by around £700. There is a wide range of accessories to support the plotter, including a choice of four types of pen in eight different colours. Further details from Apple Computer (U.K.) Ltd, Eastman Way, Hemel Hempstead, Hertfordshire. Telephone: (0442) 60244. 



(continued on next page)

A plotter to keep you out of the red



BMC X-Y Plotter B-1000

The cost of multi-colour graphics can come pretty high these days and some new plotters simply put you into the red with very little in the way of power for money to show for it.

Now Encotel Systems bring you a plotter of the quality you would expect from a company like BMC – at a price you most certainly would not.

The new BMC X-Y Plotter B-1000 offers you a short cut to business efficiency at the most attractive cost-to-performance ratio you will find anywhere. From Encotel, the price starts at just £745.

Four colour plotting on a generous A3 area is backed up by a versatile graphics ROM package option with a choice of three types of bar graphs, line graphs or pie charts.

Compact and lightweight, it can be fitted into any desk space and easily moved around the office.

The BMC X-Y Plotter simply plugs into any microcomputer with a centronics or RS232C interface including Apple, Sirius, IBM, TeleVideo, Future Computers and Superbrain.

ENCOTEL WEST

77 Laleham Road,
Staines,
Middlesex.
TN18 2EA
Telephone: Staines
(0784) 63466
Telex: 932905 LARCH G

ENCOTEL NORTH

Lockside Mill,
St. Martins Road,
Stockport,
Cheshire
Telephone: 061 449 0431

SYSTEMS
ENCOTEL

ENCOTEL SYSTEMS LTD.

7 Imperial Way,
Croydon Airport
Industrial Estate,
Croydon,
Surrey CR0 4RR
Telephone: 01 686 9687
01 680 6040
Telex: 8951921 ENCO G

News: hardware

(continued from previous page)

Legend helps Vic-20's bad memory

THE VIC-20 is now good value at around £100. But the one thing the machine lacks is memory, only having about 3.5K give or take a byte. To plug this gap comes a 27K memory board from Legend Valley Computer Systems of Newark, Ohio. One advantage of the system is that the board fits inside the Vic's case, and does not sit precariously at the back like most others. The boards cost \$129.95

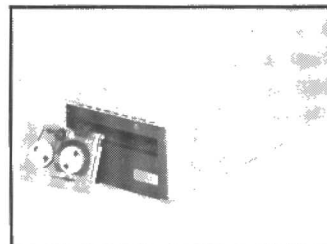
Just in case you are curious about this sort of thing, Legend Valley is the ancestral home of the Hopewell Indians who now sit around all day building peripherals with their tribal soldering irons. For more details contact Legend Valley Computer Systems, 1474, Newark, Ohio 43055.

Tape back-up for Hal Winchesters on Apples

THE WINSTREAM 20 is a tape back-up device for users of the Hal Winchester discs on the Apple range of computers. It is capable of handling 20Mbyte and is a fast and effective security device, allowing the hard-disc user to make a copy of

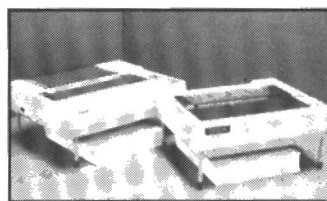
his valuable software and data.

It will also work in conjunction with the Apple Profile disc system and, if used with the right interface, a wide range of other microcomputer systems. For further details contact Stuart Hamilton, Hal Computers Ltd, Invincible Road, Farnborough, Hampshire. GU14 7QU. Telephone: (0252) 517171.



Epson on stilts

With Stilts attached to your Epson printer you can put a 3in. high stack of paper underneath. Stilts cost £3.95 for the MX-80 or FX-80, or £4.45 for the MX-199. Details from Pete and Pam Computers. Telephone: Rossendale (0706) 212321.

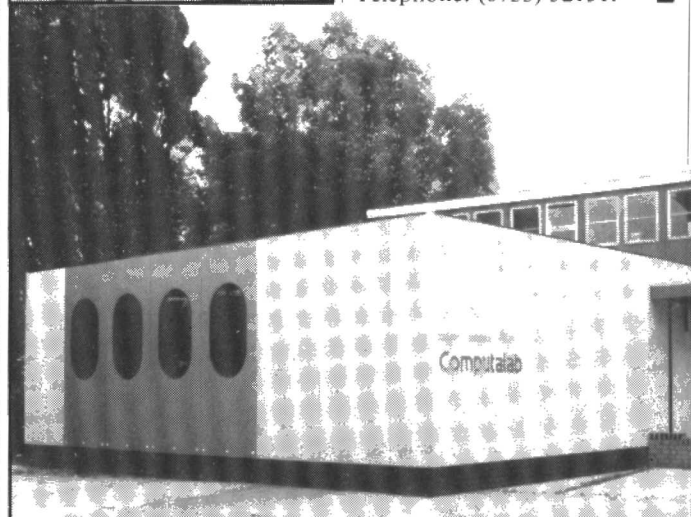


The complete computer system

Computalab is just the thing for the rich parent teachers associations to buy. It contains 12 BBC Model B-based workstation with Econet interfaces and complete provision for 24 children, right down to 24 black-coated aluminium hat

and coat hooks. There are two large wall-mounted colour monitors for classroom demonstrations and a video cassette recorder for playing back educational TV programs.

Teacher has another BBC Model B along with a disc-based Acorn System 5 file server and an Olivetti ink-jet printer. The price, including air-conditioned building, is £38,000. Details from Elliot-Medway Ltd, Glebe Court, Peterborough PE2 8EE. Telephone: (0733) 52151.



Apple Prodos

Apple are releasing a new operating system for the Apple II and IIe called Prodos. Aimed at the professional software developer Prodos resembles SOS, Sophisticated Operating System, as used on the Apple III. Prodos uses the same data formats as SOS and provides a similar Unix-like hierarchical file structure.

Apple DOS will continue to be the standard Apple II operating system, but by releasing Prodos Apple are responding to the need for a

better development environment. Apple say Prodos allows larger file sizes, more efficient memory-management, better response times, and that it makes disc-based applications device-independent.

Prodos will not be on general retail sale until early 1984, but it is available now to software developers under licence. For details contact Apple Computer (U.K.) Ltd., Eastman Way, Hemel Hempstead, Hertfordshire HP2 7HQ. Telephone: (0442) 60244.

Micro replaces maths teacher

Fun Mathematics on Your Microcomputer, is by Czes Kosnowski. The book discusses mathematical principles with lots of program examples and games written in a non-machine specific Basic. Published by Cambridge University press at £4.95, ISBN 0 521 274 516.

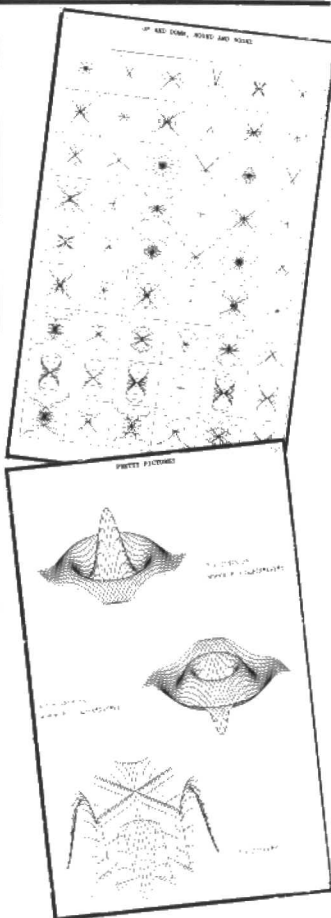
Olivetti comes in from the cold

MS-DOS can now be obtained along with CP/M-86 for the M-20, Olivetti's heavily promoted 16-bit computer. The two operating systems come together with an 8086 add-on processor card for a price of £200. With the card fitted the M-20 should be able to read IBM PC formatted discs. The MS-DOS is MS-DOS version 1.

The M-20 has until recently been out on something of a software limb. The system is built around the rather unusual Z-8000 processor chip and comes supplied with an Olivetti own-brand operating system. The new processor card turns it into a more conventional machine costing, with the 8086 card fitted, £2,695 for a system with twin floppy drives. Contact: British Olivetti Ltd, 86-88 Upper Richmond Road, London SW15 2UR. Telephone: 01-785 6666.

Last One cheap on Commodore 64

The latest serious software product to become available for the Commodore 64 is DJ 'AI'



Systems well publicised program generator, The Last One. At £85, the price is lower than versions of the product for other machines, in line with the lower price of the 64 which doubles as a home entertainment machine.

DJ 'AI' Systems has also just released The Last One for the Zenith Z-100 and the Hitachi MB-16001 16-bit machines, this time at the more usual price of £330. Details from DJ 'AI' Systems, Station Road, Ilminster, Somerset TA19 9BQ. Telephone: (04605) 4117.

Ffosswriter

Ffoss's Correspondent Word Processing package for the HX-20, the development of which we described in *Practical Computing*, March 1983 is now on sale. The name has been changed to Ffosswriter because of a name clash with another product, but it is the same ambitious package as described in the "Computing on the Train" feature.

What distinguishes it most from other text-editing packages for the Epson is its disc-like random access handling of the HX-20's microcassette drive. It allows the user to operate conveniently with named documents and makes block copying operations between different documents possible.

Supplied as a plug-in EPROM along with a 50-page reference manual, Ffosswriter runs on the HX-20 with or without the expansion unit fitted, and costs £95. Full details from Ffoss Ltd, 112 Bath Road, Slough SL1 3SZ. Telephone: (0753) 820277.

Apple card

Advanced Logic Systems' CP/M Plus card for the Apple II and Apple IIe is now available in the U.K. At £300

the plug-in processor card is good value, it includes not only the new CP/M Plus operating system from Digital Research but also an extra 64K of RAM, CBasic, GSX Graphics, and various software utilities. The card uses the fast 6MHz version of the Z-80 processor chip. Contact Scope Systems, 13 Carlisle Road, Queens Park, London NW6 6TL. Telephone: 01-969 9365.

Lots of art for the BBC

BBC computer owners have two new drawing packages to choose from. Beeb-Art which is from Quicksilver lets you draw lines or shapes in any of the 16 Mode-2 colours and save them to cassette. It costs £14.95 and comes on cassette for the BBC Model B, with or without joysticks.

Easy Graphics from Hexagon Software is a similar package additionally featuring rubber-band line drawing. Costing £13.50, Easy Graphics also comes on cassette and runs on either a Model A or B machine with at least 32K of RAM. Joysticks are not required.

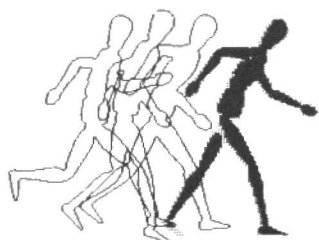
More details from Quicksilver Ltd, Palmerston Park House,

(continued on page 24)



BCPL, Forth and Lisp — three of the languages with the biggest cult followings — are now available for the BBC computer. BCPL is a structured language widely used in universities as an alternative to assembler. Forth is becoming increasingly well known for producing fast, compact code, and is ideal for machine control and graphics applications. Lisp is a list oriented language much used for artificial intelligence research and writing expert systems. Lisp and Forth are available on either cassette at £16.85, or on disc at £19.90. The user guides cost £7.50 each. BCPL is more expensive at £99.95 for a pack containing the run-time system on ROM along with other parts of the system on disc and the user guide. The BCPL user guide costs £15.50 bought separately. More details from Acornsoft Ltd, 4A Market Hill, Cambridge CB2 3NJ. Telephone: (0223) 316040.

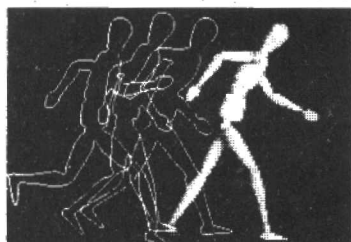
GRAPHICS - DATA CAPTURE - MEDICAL PHYSICS - VIEWDATA -
SECURITY - FACSIMILE - DESIGN - PRINTING - BADGES -
CONSUMER TEXTILES - ILLUSTRATION



We captured this little manikin running to see a demonstration of the versatile

DIPLOMAT VIDEO DIGITIZER

£195 (Apple II+)
£345 (Apple //e, includes 64K Extended 80 Column Card)
Plus Carriage and VAT



Details and demonstrations at PCW Show, Stand 345, Hall B, Lower Level (Pete & Pam) and COMPEC '83, Stand 718, National Hall (CompuTech), your nearest Apple Dealer, or the manufacturer:

COMPU TECH

SYSTEMS

168 Finchley Road London NW3 6HP
Tel: 01-794 0202
Telex: 268048 EXTLDN G

● Circle No. 114

Want reliable power for today's electronics without costly special wiring?



Sola Minicomputer Regulators and Mini UPS systems replace the dedicated line plus provide greater protection against power line disturbances.

FOR MORE INFORMATION ABOUT THE SOLA POWER PROTECTORS CONTACT:

28 LURKE STREET
BEDFORD MK40 3HU, ENGLAND
PHONE 0234 40094
TELEX B26431

A UNIT OF GENERAL SIGNALS
SOLA-BANNER
(EUROPE) LTD

● Circle No. 115

News: software

(continued from page 21)

13 Palmerston Road,
Southampton SO1 1LL.
Telephone: (0703) 20169. And
from Hexagon Software, 17
Straits Road, Gornal, Dudley,
West Midlands DY3 2UR.
Telephone: (0384) 232992. □

Sharp and Tandy statistical forecast

Easi-Trend for the Sharp PC-1500 and Tandy PC-2 pocket computers enables users to identify trends and make forecasts from entered data. The program comes with a manual explaining statistical forecasting, and costs £19.95, including VAT. For more details contact Elkan Electronics, 11 Bury Road, Prestwich, Manchester M25 9JZ. Tel: 061-798 7613. □

Specific packages for Commodore

Specific Software has released a range of tape and disc-based programs for the Vic-20 and Commodore 64 to do invoicing and sales and purchase accounts. Specific say the disc versions can handle 300 accounts and up to 2,000 transactions, while the cassette

versions are good for 60 accounts and 300 transactions.

Prices range from £20 for a Vic-20 invoicing program to £150 for disc-based sales accounts with integrated invoicing for the Commodore 64. Details from Specific Software Ltd, 10 Farlands Road, Stourbridge, West Midlands DY8 2DD. Telephone: (03843) 73377. □

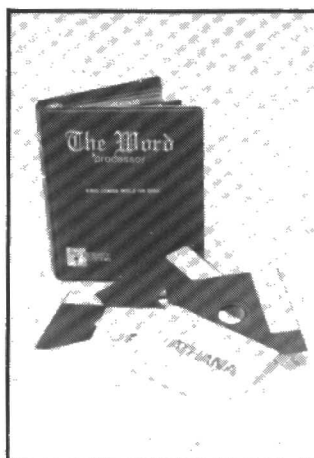
Image analysis system on ACT Sirius

Digithurst's image-analysis system will now work with the ACT Sirius computer. The Microsight image-capture package consisting of video camera, interface box and software achieves a resolution of 256 by 256 pixels and costs £495. The Microscale software suite consisting of programs to manipulate images and measure perimeters and areas costs £295. Versions of the system are available for other micros including the BBC and Pet computers. Contact Digithurst Ltd, Leaden Hill, Orwell, Royston, Hertfordshire SG8 5QH. Telephone: (0223) 208926. □

Hewlett-Packard integrated package

MBA, the integrated spreadsheet, graphics and filing package from Context can now be obtained for Hewlett-Packard's new 16-bit 68000-based machine. Context MBA's core function is a spreadsheet of 95 columns by 999 rows. The user can enter text, numbers and formulae in any cell. As any cell can hold up to 8,000 characters MBA can be used as a database and as a simple word processor. The package allows you to do sorting and searching operations and to construct several different kinds of graphs from cell data.

Context MBA is already available for the IBM PC, requiring the 256K RAM expanded system to run. The standard HP Series 200 model 16 comes with half a megabyte of RAM and uses the powerful 68000 processor, so it is well suited for this type of large,



The new software package called The Word processor is not quite what it seems. It is the King James Bible on disc. The complete text is contained on a set of discs along with a program which lets you search the scriptures for any word or phrase you wish to refer to. Apple and IBM PC versions are available from Pete and Pam Computers at £149. Contact Pete and Pam Computers, New Hall Hey Road, Rossendale, Lancashire BB4 6JG. Telephone: (0706) 212321. □

Software dealer to join Softsel

Software dealer SBD Software is to gradually stop trading. Susan Ben-David, who owns the company, is closing it down and joining Softsel, the large American software distributor which has recently set up an operation in this country. Susan Ben-David's job as product services manager includes the task of selecting British and European software for distribution by Softsel worldwide. 

More packages for BBC accounting

Six disc-based business packages for the BBC Computer have been announced by HCCS, covering nominal, purchase and sales ledgers, stock control, order processing and payroll. Each package costs £59.95 and can be used on its own or as a module integrated with the other packages in the range. Details from Home and Continental Computer Services Ltd, 22 Market Square, Biggleswade, Bedfordshire SG18 8AS. Telephone: (0767) 317300. 



multi-function general package. The HP keyboard's unusual cursor-control knob can be used to scroll around inside Context MBA.

Context MBA for the HP system costs £593. A Model 16 system with twin Sony microdrives costs £5,213. Details from Personal Computer Literature Department, Hewlett-Packard Ltd, Customer Service Centre, Winnersh, Wokingham, Berkshire RG11 5DZ. Telephone: Crowthorne (0344) 773100. 

Which home computer gives you so much software for so little?

Entertainment for all:

Video Chess	£34.95
Munchman	£29.95
Parsec (Optional Speech)	£29.95
Othello	£24.95
Tunnels of Doom	£24.95
Adventure Cartridge	£24.95

Additional Adventure Games:
all at £14.95

Adventure Land	
Mission Impossible	
Voodoo Castle	
The Count	
Strange Odyssey	
Mystery Fun House	
Pyramid of Doom	
Ghost Town	
Savage Island	
Golden Voyage	

Alpiner	
(Optional Speech)	£24.95
TI Invaders	£19.95
Car Wars	£19.95
Chisholm Trail	£19.95
Hustle	£14.95
Tombstone City	£14.95
Connect Four	£14.95
Video Games 1	£14.95
Hunt The Wumpus	£14.95
Five-A-Side Soccer	£14.95
Amazing	£14.95
Attack	£14.95
Blasto	£11.95
Blackjack & Poker	£11.95
Zero Zap	£11.95
Yahtzee	£11.95
Oldies But Goodies 1	£11.95
Oldies But Goodies 2	£11.95
Market Simulation	£11.95

Educational:

Early Reading	£29.95
---------------	--------

Music Maker	£29.95
Alligator Mix	£24.95
Alien Addition	£24.95
Demolition Division	£24.95
Dragon Mix	£24.95
Minus Mission	£24.95
Meteor Multiplication	£24.95
Touch Typing Tutor	£24.95
Addition Subtraction 1	£19.95
Addition Subtraction 2	£19.95
Multiplication Division	£19.95
Numeration 1	£19.95
Numeration 2	£19.95
Early Learning Fun	£14.95
Beginning Grammar	£14.95
Number Magic	£14.95
Hangman	£11.95
Teach Yourself Extended BASIC	£11.95
Beginners BASIC Tutor	£9.95

Home Organisation:

Personal Record Keeping	£39.95
Personal Report Generator	£39.95
Home Budget	£24.95
Home Financial Decisions	£24.95
Personal Financial Aids	£11.95

Other Applications:

Inventory	£85.95
Invoicing	£85.95
Mailing List	£85.95
TI Writer (Word Processing)	£85.95
Multiplan (Spread Sheet Program)	£85.95
Statistics	£54.95
Terminal Emulator	£49.95
Maths Routine Library	£29.95
Electrical Engineering Library	£29.95

Graphing Package	£29.95
Structural Engineering Library	£29.95
Programming Aids 2	£19.95
Programming Aids 3	£19.95
Speech Editor	£19.95
Programming Aids 1	£11.95

Programming Languages:

PASCAL Editor	£99.95
PASCAL Linker	£79.95
Extended BASIC	£69.95
TI Logo	£69.95
Editor/Assembler	£69.95
Mini Memory	£69.95
PASCAL Compiler	£59.95

A.S.K. Applied Systems

Knowledge:

Hide and Seek	t.b.a.
Number Gulper	t.b.a.

Collins-Educational:

TI-99/4A Starter Pack 1	£9.95
TI-99/4A Starter Pack 2	£9.95
TI-99/4A Game Writer Pack 1	£9.95
TI-99/4A Game Writer Pack 2	£9.95
Chess Learner Pack	£9.95
Record Keeper Pack	£9.95

Ivan Berg Software:

Maths Tester 1	£9.95
Maths Tester 2	£9.95
Physics Tester	£9.95
Chemistry Tester	£9.95
Biology Tester	£9.95
Human Biology Tester	£9.95

Bond Associates:

Easycalc	£49.95
----------	--------

Little Genius Ltd:

Scrabble	£29.95
----------	--------

And is available at all these dealers?

All branches of:

Argos, Comet, Dixons, Greens of Debenhams, John Lewis, Rumbelows, Wigfalls, Zappo.

Major branches of:

Asda, Binns, Computers for All, CO-OP, Currys, Fine Fare, Ketts, Photomarket, Rymans, Spectrum, Telefusion.

And at:

ABC Computers - St Austell
Akhter - Harlow
Anglia Audio - Bedford
Anglia Sound - Stevenage
Audio Marketing - London
Audio Vision - Faversham
Bagnall - Stafford

Carvells - Rugby
Combined Trading - Hatfield
Computer Supermarket - Manchester
Cotton TV - Peterborough
Dean and Son - London
Delta Electronics - London
Densham Computers - Poole
Desk Aids - Southampton
Dodar - Ashton
Fenwicks - Newcastle
Galaxy Video - Maidstone
Hamleys - London
Harrods - London
Heffers - Cambridge
Hyman Computers - Manchester
Landau - Sutton
Lion House - London

Micro Value - Amersham, Bucks.
Midshires - Crewe
Milequip - Gloucester
OEM Computers - Rugby
Parco Electronics - Honiton
REW - London
Robox - Glasgow
Science Studio - Oxford
Selfridges - London
Star Trek Video - Wigan
Toy and Hobby - Wigan
Universal Warehouse - Reading
Video Palace - London
Vision Store - Kingston
Welwyn Dept. Store - Welwyn
And many other leading Computer Stores.

Turn the page and see the unbeatable.

"At last in its proper setting, the infinite has assumed a respectable place next to the finite, just as real and just as dependable, even though wholly different in character. Whatever the infinite may be, it is no longer a purple cow."

Edward Kasner and James Newman
Mathematics and the Imagination, 1940

IN GÖDEL'S arithmetisation of mathematics, the key concept is that of the Gödel number. Hilbert, in 1904, had noted that symbolic logic could be treated as though it were a branch of elementary number. But it was Gödel, in 1931, who worked what this actually implied in practice. The method was directed towards Russell and Whitehead's *Principia Mathematica*, and an English translation of his original German title might be "on formally undecidable sentences of *principia mathematica* and related systems."

Gödel starts with a set of basic axioms of number theory which effectively correspond to three of Peano's axioms of number:

- the successor of any number cannot be zero;
- if the successors of two numbers are equal, then the two numbers are also equal;
- and if a certain property is true for the number zero, and if true for any number it is true for its successor, then the property is true for all numbers. The latter is the axiom of mathematical induction.

In Gödel's logical symbolism, the number zero is shown as 0, the number one is shown by f0, the successor of 0, two is shown by ff0, and so on. There is only one primitive number in his system and that is 0, all other numbers are the results of operations on that number using the primitive f. Other primitives are simple variables, for example, X in his first axiom

$$\sim (fX = 0)$$

it is not true that the successor of any number is zero. These primitives are used to create secondaries, more complex arrangements which on analysis end up being assertions about numbers or variables.

At a different level there are secondaries which involve propositional expressions, or expressions which can be turned into such, and assertions about the relationship between the elements implicated. For example, in his second axiom

$$fX = fY \rightarrow X = Y$$

if the successor of X is equal to the successor of Y, then X is equal to Y. Either side of the implication is ultimately composed of primitives but the implication is of a different order of things.

In the 1931 article Gödel starts his arithmetisation by associating each of the primitive signs in his symbolism with a natural number.

Symbolic logic

Boris Allan continues his analysis of Gödel's work in relation to threaded interpretive languages.

Symbolism and natural numbers.	0	1
	f	3
	~	5
	v	7
	P	11
	(	13
	)	17
	X	19
	Y	23

and so forth — any scheme of assignment which uses the prime numbers would obviously do. The Gödel numbers for these primitives correspond to the addresses of the locations for primitives in threaded interpretive languages. If you show a property of X, that is a higher order type, by X2 then this is given the Gödel number 19². A property of a property of X is X3, with a Gödel number 19³, and so on until infinity.

A secondary in a TIL or threaded interpretive language also has an address, that is, a number, and that number when used points to an unambiguous set of further addresses, which are either the addresses of primitives or further secondaries. The Gödel numbers of primitives are associated in a special way to enable the content of any formula to be established from the Gödel number of the formula.

The third of Gödel's axioms is $X2(0).X P(X2(X)) \rightarrow X2(fX) \rightarrow X P(X2(X))$ that is, if there exists a property of X which is true for 0, and if, for all X, when true for X it is true for the successor of X, then the property is true for all X. Note that mathematical induction cannot be expressed in a single axiom without use of a variable of a higher type than is X2.

The implication $\rightarrow$ is not one of Gödel's set of primitive symbols, but as $A \rightarrow B$ is the same as $\sim A \vee B$, part of the above can be re-written

$$X P(\sim X2(X) \vee X2(fX))$$

which has the Gödel number $2^2 19^* 3^2 11^* 5^2 13^* 7^2 5^* 11^2 19^2 2^*$... * 47¹⁷

This number, though large, can be unambiguously factorised into its constituent elements so that you can

always reproduce the formula. If the numbers of formulae in a proof, a proof is no more than a sequence of logical formulae, are F1, F2, F3, ... Fn, then the Gödel number for the proof is

$$2^2 F1 * 3^2 F2 * 4^2 F3 * 5^2 F4 \dots$$

and this method associates one and only one number with each formula or sequence of formulae. This is the arithmetisation of mathematics — a bootstrapping exercise.

The threading through addresses which characterises TILs has a very close analogue here. You have a Gödel number which is factorised at the first level; you have a TIL word which produces a series of addresses; some or all of the numbers which arise from the factorisation have then to be factorised to produce further numbers; some of the addresses lead to sets of further addresses. The process continues, on both accounts, until you reach the system primitives.

The transfinite numbers are shown by the Hebrew for A, aleph, but to save typographical contortions I will simply use A — though still calling it aleph. A few characteristics of the first transfinite number aleph-null, A(0):

$$\begin{aligned} A(0) &= A(0) + 1 \\ A(0) &= A(0) + A(0) \\ A(0) &= A(0) * A(0) \end{aligned}$$

though

$$A(1) = A(0)^A(0)$$

where A(1) is the next transfinite number, aleph-one. If these strange equalities are studied, it is obvious that they contradict Peano's axioms, for one property of zero is that it is different from its successor. So is it always possible to unambiguously factorise a Gödel number, is there some point at which the number is "too large"?

Return to the TIL. There comes a point at which the physical confines of the computer memory mean that we cannot extend the words in the memory any further. A TIL is manageable, it respects finity and it also asks for the mechanism by which an operation is to be performed. A word may, for example, refer to itself — a TIL asks what that self-reference means in practice.

Any secondary in Gödel's scheme will
(continued on page 38)

So cheap to keep in touch

THE COMPUTING power now available to the individual user via the new generation of 16-bit microprocessor-based personal computers is quite phenomenal. In many respects it exceeds the capabilities offered by those big expensive mainframe systems, which were once the mainstay of the computer industry. But despite their obvious power, most personal computers still lack one of the most useful features of the mainframes: easy communication with other users and the ability to share a common database.

However, judging by the latest batch of microprocessor peripheral devices this is a drawback which will soon disappear. Before long it will be the exception rather than the rule for personal computers to operate in splendid isolation in a dark corner of the office or living room.

Many multi-user systems particularly, need to communicate over long distances by the public telephone system. This need is currently satisfied by Modem — modulator/demodulator — units, which can be used to convert the serial RS-232 line of a terminal or VDU to the voice band signals expected by the telephone network. Trouble is these Modems are quite complex as they have to convert the logic-level signals from a terminal into audio frequency tones before transmission; they do the reverse at the receive end. They must also cope with the distortions inherent in long distance connections and the many other peculiarities of the telephone link, which may include landlines, microwave links and even a satellite between the transmit and receive terminals. Complexity equals expense and so this form of communication has in the past been largely ruled out for low-cost personal-computer applications.

But the semiconductor chip manufacturers have not been neglecting this problem. Now that the personal-computer revolution is well underway advances in chip technology and the attractions of a huge potential market have spurred the development of self contained Modem devices. These can be built into every microcomputer at very low cost to provide a direct telephone connection to even the most humble office system.

The integration of a complete Modem system onto a single silicon chip is by no

means a trivial matter. Traditional Modems are essentially analogue — rather than digital — systems, and rely heavily on the use of sine-wave oscillators and inductor filter circuits for correct operation. To satisfy the single-chip requirement an analogue signal is simulated using digital techniques, with the result that the new generation of Modems are really high-speed dedicated microprocessor systems, internally as complex as the 16-bit general purpose microprocessors they will support.

Several of the major chip manufacturers have recently announced sophisticated

by Ray Coles

single-chip modems, which will ensure that competition is fierce and prices low; the one which caught my eye was the AM-7910 from Advanced Micro Devices.

AMD has brought the traditional advantages of VLSI digital circuitry to bear on the problem and has made its device totally programmable in order to suit the various standard communication protocols in use in the U.S. and Europe. Analogue-signal generation and processing is simulated by using a high-speed digital-signal processor, which has its own 24K ROM, 1.3K RAM array, digital-to-analogue and analogue-to-digital converters fabricated on the same chip.

The entire system lives in a tiny 28-pin dual-in-line package and runs from dual 5V supply rails using just 600mW of power. Voice band Frequency Shift Keying, FSK, data rates of 300, 600 and 1200 baud can be selected, as can one of the nine Bell and CCITT recommended communications protocols.

With this sort of capability now available for a few pounds, we can expect all future microcomputer systems to have long distance communication facilities available as a standard feature. This would allow even a basic office micro to keep in close touch with all that lovely data available in the outside world.

Modem links are good for long distance access to a central data base or larger computer, but due to the limited frequency response of the standard telephone network data rates are restricted, making the transfer of large quantities of data a tedious

business. Over shorter distances data transfer rates can be increased dramatically by avoiding the restrictions of the telephone system; using instead dedicated high-speed communications links called Local Area Networks, LANs.

Using a LAN, such as Ethernet, data transfer rates of 10 million bits per second are possible. This means all the microcomputers in, say, an office block can be linked together for the interchange of messages and the sharing of precious resources like hard-disc systems and line printers.

Unfortunately LAN controllers are complex and therefore expensive. But the semiconductor manufacturers are falling over themselves to provide cheap VLSI solutions, and a whole flood of new devices are about to be launched into an eager market.

Ethernet controllers are a good deal more complex than the simpler Modems, but their complexity is more easily handled using digital techniques. Again the new generation of single-chip controllers will depend heavily on the use of dedicated microprocessors to provide the clever bits.

Take the Intel 82586 LAN controller: when used with the companion 82501 driver chip, required to drive the coaxial cable used for interconnection, the device will implement the full Ethernet specification as defined by the original sponsors of the standard, the DEC, Xerox and Intel grouping. In the past about 80 integrated circuits have been required for the job, but with the advent of the 82586 a single 48-pin package is all that is needed.

The new Intel device takes the burden of link control away from the associated microprocessor. It merely requires it to assemble a message for transmission in its own memory space, or to retrieve received messages placed back in the microprocessor memory space by the controller. The 82586 has a built in DMA controller which allows it to take control of the system bus for the retrieval and replacement of messages, only interrupting the busy CPU when all the hard work has been completed.

Eventually we can expect LAN controller chips like the 82586 to cost less than £20 each, making the provision of this form of communication a logical option for future 16-bit machines. □

Symbolic Logic

(continued from page 29)

be shown as Sec? where the ? is replaced by an identifier — compare my earlier analysis of secondaries for TILs. Each secondary, say a formula, will be given a Gödel number Number.Sec?, and to find what the number means, that is, to unravel the formula, you have to factorise that number. To factorise the Number.Sec? to find what is the formula you

NUMBER.SEC? EXECUTE
in direct analogy to a TIL.

Take a secondary SecX, and suppose this is composed of a series of other secondaries, taken in order, SecA, SecB, and SecC

: SECX SECA SECB SECC ;

or

: SECX NUMBER.SECA EXECUTE
NUMBER.SECB EXECUTE
NUMBER.SECC EXECUTE ;

where Number.SecX is the Gödel number of SecX, and likewise for the others. How is it possible to incorporate variables? A variable is effectively a dummy which can be replaced by any value; it is an Inputnumber, in terms of my earlier analysis of TILs. You will assume that

Inputnumber.1 is the first variable, and it may appear more than once in a definition, and the same for Inputnumber.2, etc.

The operation ?Provable when applied to a number gives the result true if the sequence of formulae are a valid proof within the system, false otherwise. That is

NUMBER.SECX ?PROVABLE

and now you can produce Gödel's famous result.

Let SecY be the main secondary in which you are interested, let SecX comprise the main body of the sequence of formulae, and let there be a variable Inputnumber, which corresponds to X in the arithmetic

: SECY SECX INPUTNUMBER EXECUTE ;
As you saw in the first part, Inputnumber can be replaced by Number.SecY, and so SecY can be re-written as

: SECY SECX NUMBER.SECY EXECUTE ;
— one form of recursion.

Gödel designs a special formula, SecG, first he makes the simple formula

: SECG INPUTNUMBER
? PROVABLE NOT ;

which asserts that the formula whose number is to be supplied, via Inputnumber, cannot be proven. The number for this formula is Number.SecG, and what Gödel does is substitute for Inputnumber:

: SECG NUMBER.SECG
? PROVABLE NOT ;

and my earlier worries about the meaning

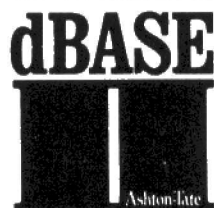
of recursion, in part 1 in the September issue, are reinforced.

Consider how a TIL might analyse the formula/definition. It would not come to any conclusion, the process would never end until memory ran out. Gödel assumes along with many other mathematicians that in mathematics the story need never end, it could carry on until infinity. However, as noted earlier, this is not to say that at some non-Peano transfinity the solution would not be resolved. People can resolve it.

Computers are not generally used to play meaningless games, apart from some reaches of AI and computer science — long may it continue. TILs were developed to provide a powerful method of using computers. Interestingly, it seems as if the TIL philosophy is a practical application of metamathematics, the arithmetisation of mathematics.

It is generally acknowledged that Gödel's method is the most powerful method yet devised for studying mathematics and logic, and this corresponds to the power of TILs. Gödel's method does have its problems, the unprovability formula, but these are only the problems inherent in the extension of the infinite of finite ideas.

A TIL is an artificial intelligence language which accepts that there is infinity but has not — as far as I know — been accepted by the AI community which still believes in the infinite.



THE RELATIONAL DATABASE SYSTEM

FOR MICROCOMPUTERS

- TRAINING?
- ADVICE, GUIDANCE, CONSULTANCY?
- PROGRAMMING?
- DATABASE 'PRIMER' MANUAL featuring 'dBASE II'

For further details, contact:

Lionel Boreham

LANTECH Information Systems Ltd.

55 Peascod St. Windsor, Berks SL4 1DE

Tel: Windsor 58182/58013

LANTECH
Information Systems

● Circle No. 122

A FLOPPY FOR YOUR MICRO

SHARP

BBC

TRS 80

**250K
£142**

**500K
£182**

**1000K
£252**

- TWELVE MONTHS GUARANTEE AVAILABLE AS ONE OR TWO DRIVE SYBSYSTEMS
- BACKED BY SERVICE COMPANY TO THE TRADE
- DRIVE REPAIRS FROM £25

SPECIFICATION		NAKED	BOXED WITH PSV
250K	40 TRACK SS	£142	£182
500K	40 TRACK DS	£182	£222
1000K	80 TRACK DS	£252	£300

PRICES DO NOT INCLUDE VAT

MATRIX

(COMPUTER ENGINEERING) LTD

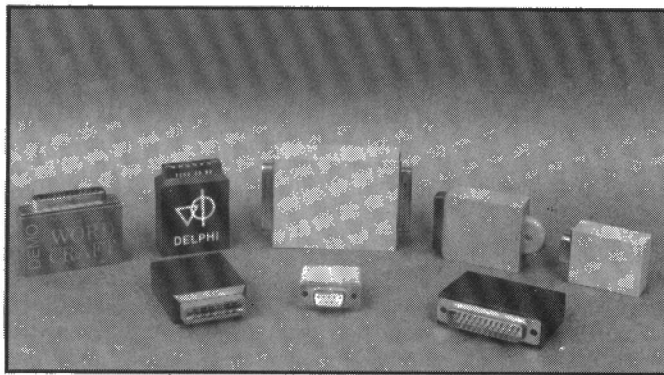
0225 742486

● Circle No. 12

WORDCRAFT SYSTEMS first developed its dongles for Commodore and, later, Sirius micros. Now it has a version for the IBM PC. It fits into the parallel printer port — providing another port for your printer — and contains code without which your software will not run. On the good side, at least the system allows you to make security copies.

PC dongles cost £15, which is somewhat more than the £2.50 for the Vic-20 version. The minimum order is 100 so you cannot get a sample from Mike Lake, Wordcraft Systems, Oak Lodge, Farley Road, Derby DE3 6BW. Telephone: (0332) 683892. 

PC dongles from Wordcraft



Maximum expansion from Legend

LEGEND has introduced an expansion card which will support up to 768K of directly addressable dynamic RAM, in either 64K or 256K segments. The card maps round the address space already used to provide the maximum possible, 1Mbyte.

Contact Legend Industries, 2,220 Scott Lake Road, Pontiac, Mi 48054. Telephone: (313) 674 0953. 

0.5 Mbyte going cheap

ENCOTEL is now importing the Profit Systems RAM expansion card, which provides an extra 512K for £445. PC-DOS 2.0 is included free.

Contact Encotel Systems, 7 Imperial Way, Croydon Airport Industrial Estate, Croydon, Surrey CR0 4RR. Telephone: 01-686 9687/8 

Addressbook

DECISION TECHNOLOGY has adapted its well-known

Addressbook program for the IBM PC, with an XT version to follow. It is said to be easy to use and costs only £90. A WordStar interface is provided for use with Mailmerge.

Contact Decision Technology, 7 St Johns Road, East Molesey, Surrey KT8 9JH. Telephone: 01-979 5533. 

Front end revelation

REVELATION is the name of a Pick look-alike front end to PC-DOS, which is useful if you have the PC linked to an IBM Series/1 super-mini running a full version of the Pick operating system. It turns the PC into a minicomputer terminal, with access to a wide range of software, while retaining the ability to run packages under MS-DOS. A new application generator, Appgen, can be run under Pick or under Unix, providing a bridge between the two systems.

Contact Interactive Data Machines. Telephone: (0302) 786677. 

Pearls of wisdom

THE SYSTEMS generator Personal Pearl has now been released in a version for the IBM PC. It not only runs under PC-DOS but also under CP/M-86 and Concurrent CP/M. Pearl is a relational database that outputs ASCII files for use in word processing and links to the Supercalc spreadsheet package. It uses

the IBM's function keys, and costs £190. Contact Pearl Software, 12 Christchurch Road, Bournemouth BH1 3LD. Telephone: (0202) 20692/3.

Graffcom has been rewriting its 8080/Z-80 machine-code packages in 8086/8 code to make full use of 16-bit CPUs. The new range, designated 2020, includes word-processing, financial-planning and Configurable Manager packages. The series has just been implemented on the IBM PC. Contact: Graffcom Systems Ltd, 102 Portland Road, London W11 4LX. Telephone: 01-385 9422.

Micropro, the publisher of WordStar, now has all its software available on the PC. The latest offerings are CalcStar — improved to offer 1,300 cells — and InfoStar WordStar, SpellStar and Mailmerge are, of course, already familiar under PC-DOS. Contact Micropro International Ltd. Telephone: 01-487 5728/9. 

The Strategist

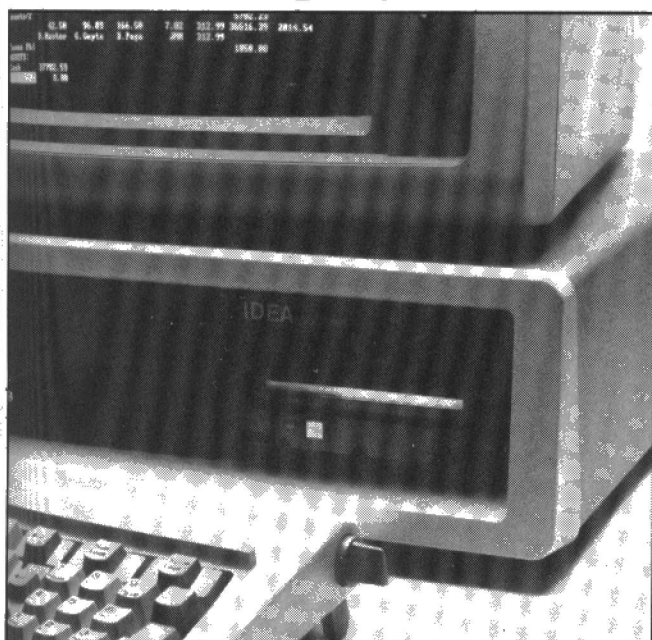
ASHTON TATE, author of dBase II, has launched a new financial package called the Strategist. After you enter 31 key business assumptions, the program tells you if your proposed project will succeed or fail. The information is presented as 44 graphs and three detailed reports. Results can be sent along to dBase II. Contact Skye Quin at Ashton Tate (U.K.) Ltd. Telephone: (0908) 568866. 

Edison on PC

THE EDISON portable software system runs on a PDP-11/23 minicomputer, and now also on the IBM PC with 256K of RAM. Edison is a Pascal-like language.

The Edison system includes an operating system, compiler, screen editor, text formatter, print program and assembler. For more information read Per Brinch Hansen's book *Programming a Personal Computer*, published by Prentice-Hall.

Contact Per Brinch Hansen, Computer Science Department, University of Southern California, Los Angeles, Ca 90089. 



KPG Hardware House is the distributor of the IDE Associates range of PC disc sub-systems. Latest products are a 3.9in. 5Mbyte removable cartridge system. Either can be fitted into a standard PC in place of an existing floppy drive, or into an external expansion unit. Installation is said to take 15 minutes. Contact KPG Hardware House, 578-586 Chiswick High Road, London W4 5RP. Telephone: 01-995 3573. 

IBM'S MAIN marketing thrust of the Personal Computer has been at presenting it as user friendly. Indeed, the twin-floppy version is easy to use compared with most previous small business micros. This is due to PC-DOS, the IBM version of Microsoft's MS-DOS, and the high quality of much PC software. While PC-DOS is no one's idea of the perfect operating system, it is easier to learn than CP/M. However, with the hard disc version of the IBM PC, the XT, and the essential PC-DOS version 2 the system moves to a higher level of difficulty.

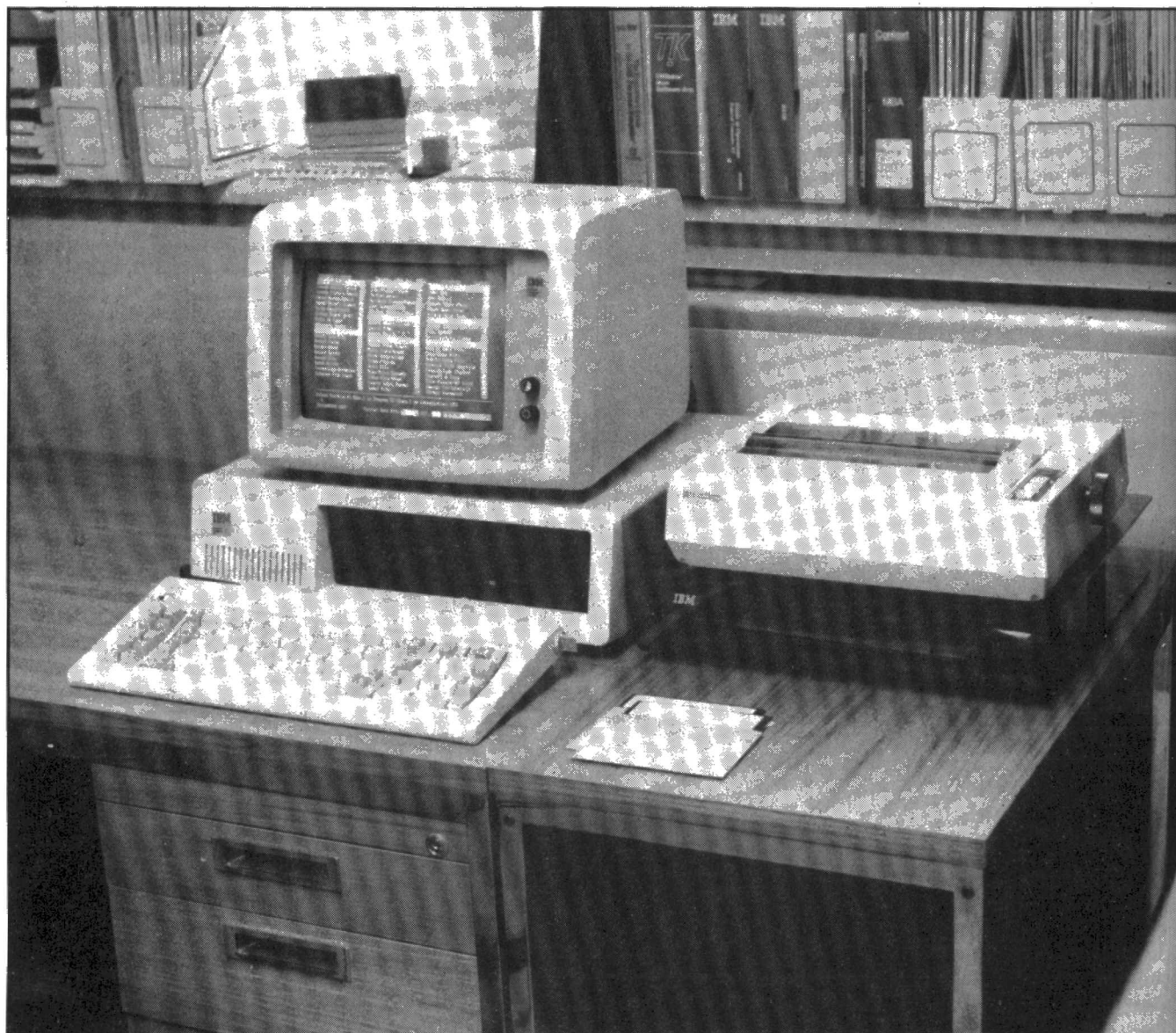
There are several reasons for this. First, DOS 2 has more commands and is inherently more complex than DOS 1; it is not just bigger. Second, controlling the hard disc requires a much more organised approach to keeping files and back-ups. Third, very little current software has been written with hard disc operation in mind. The result is that the IBM XT is currently a much less viable option for the newcomer to computing. At the very least, the typical XT user will require a higher level of dealer support, and greater personal commitment.

IBM XT

In the third part of our review, Jack Schofield looks at the operating system, which may not be so easy to get used to. The hard disc and PC-DOS version 2 demand a high level of literacy.

This is not because of the hard disc itself, which is big, fast and in principle works just like a floppy. It also takes up the same amount of room but it has two platters, giving four surfaces for data storage. Each surface has 306 tracks of 17 sectors, compared to 40 or 80 tracks on a single- or double-sided 5.25in. floppy. The total storage is thus about 10Mbytes, which is the equivalent of 32 of the 320K standard floppy discs. Access time, the time to read an item of data, is up to 10 times quicker.

IBM does not disclose the manufacture of the fitted hard disc; likely sources are Seagate Technology of California and Miniscribe Corporation of Colorado. When the XT goes into production in Scotland in November there may be an opening for a Scottish disc. The disc in the XT supplied for review offered initial formatted storage of 10,592,256 bytes or 10,344K. Even after copying on the DOS system files there was more than 10Mbyte free. Such information is very simple to



discover
DOS, w
and IO.
The m
Assign,
For, Shi
Chdir, J
Set, Tre
four nev
some of
been en
disc ope
Backu
contents
floppy
kind of
this. Th
A:/S ba
those hi
a note in
a file, s
back up
modifie
DOS als
the \D
only the
That wi
the date
In all ca
back or
respects
usual C
In ge
using I
language
The use
of pro
abbrevi
create a
though
keyboar
Submit
create A
enable p
can be r
Echo C
made in
using th
Cls to
program
system
debuggi
Most
connect
structur
made to
Microse
to divid
director
for Ma
contain
turn co
as man
limitatio
than 63
You
change
ChDir f
the roo
director
identifi
sub-dire

discover using the ChkDsk utility from DOS, which lists hidden files like DOS.SYS and IO.SYS separately.

The main new commands in DOS 2 are Assign, Backup, Break, CIs, Ctty, Echo, If, For, Shift, Goto, Graphics, Mkdir, Rmdir, Chdir, Path, Prompt, Recover, Restore, Set, Tree, Ver, Verify and Vol. There are four new characters, <, >, | and \. Also some of the existing DOS commands have been enhanced, mainly to cater for hard disc operation.

Backup has been added to allow the contents of the hard disc to be copied to floppy discs, since IBM do not have any kind of tape streamer or cartridge to do this. The simple command Backup C:\A:\S backs up all the files on C, including those hidden in subdirectories. DOS makes a note in the directory whenever it writes to a file, so the \M parameter is provided to back up only those files which have been modified since the last back up was done. DOS also keeps time and date records, so the \D parameter is provided to back up only those files created after a certain date. That will make you wish you had entered the date every day when booting PC-DOS. In all cases, Restore is used to copy the files back onto the hard disc. In most other respects Backup seems to work like the usual Copy command.

In general the new commands make using DOS 2 much more like using a language than using an operating system. The user is involved in numerous little bits of programming using Copy.Con, an abbreviation of copy from console, to create a file which sends commands just as though they had been typed in at the keyboard. Thus it is the equivalent of Submit in CP/M. Copy.Con is used to create Autoexec and other batch files which enable programs to be customised, so they can be run by inexperienced users. With the Echo Off command the process can be made invisible. Variables can be included using the % sign. Using Goto, For, To, and CIs to clear the screen, it is not unlike programming in Basic except that the system provides virtually no help with debugging.

Most of the other new commands are connected with the provision of tree-structured files, through which DOS 2 is made to resemble Unix — specifically, the Microsoft version called Xenix. The idea is to divide the hard disc into a series of directories, created by typing Md or Mkdir for Make Directory. This directory then contains files or sub-directories, which in turn contain files, and so on down through as many levels as you require. The only limitation is that the Path must not be more than 63 characters long.

You start in the root directory but can change to a subdirectory by typing Cd or ChDir for change directory. Typing Dir at the root level lists only files in the root directory and sub-directories, which are identified by <DIR>. Typing Dir inside a sub-directory lists only files in that

```
C>COPY CON:LOGON.BAT
ECHO OFF
CLS
ECHO YOUR PASSWORD IS BEING CHECKED
IF %1==JACK GOTO A
ECHO ACCESS DENIED. GET LOST!
ECHO OFF
GOTO END
:A
ECHO PASS, FRIEND
ECHO ENTER YOUR COMMAND
ECHO A = WORD PROCESSING
ECHO B = MULTIPLAN
ECHO C = STRIP POKER
:END
^Z

1 File(s) copied
```

Listing 1. A PC-DOS program which shows how you might write password system. Typing Logon Jack offers a selection of programs in a menu, whereas Logon Fred results in a Get Lost message. With more users the variable %1 would have to be compared with other possible entries. Also, it would have to be an Autoexec.Bat file and not send unwanted users straight into the system at :End, but this is just for illustration. Note that three more files have to be created, A.Bat, B.Bat and C.Bat, to run the programs from the menu. Listing 2 changes the directory to MP/JACK and runs MultiPlan, MP.

directory and the names of sub-sub-directories. The root directory is then effectively invisible to the system. The particular directories and files in use can be specified by the Path command. Thus it is simple to set up a password system where the password takes users only to their own set of files, so several different users could use the same machine.

For example, the root directory could contain half a dozen .Bat files for main applications such as word processing, financial planning, etc. Selecting one from a list Echoed to the screen, then typing a name or password could take the user into a directory containing only their own files for

(continued on next page)

```
C>COPY CON:B.BAT
CD\MP\%1
PATH\MP
CD
MP
^Z

1 File(s) copied
```

Listing 2. This routine changes the directory to MP/JACK and runs Multiplan, MP.

Specification

SYSTEM

CPU: Intel 8088 HMOS pseudo 16-bit running at 4.77MHz

Memory: 128K of RAM expandable to 640K; 40K of ROM with socket for expansion to 48K

Discs: single 5.25in. mini-floppy with 360K of formatted storage, plus 10Mbyte Winchester hard disc

Features: 62-pin expansion slots for six full and two short expansion cards, but four slots are required to run basic system.

Interfaces: cards for mono display/parallel printer and asynchronous

communications supplied as standard

Dimensions: 500mm. x 410mm. x 124mm.

DISPLAY

Type: 11.5in. green phosphor screen with brightness and contrast controls

Display: 25 lines by 80 characters

Dimensions: 380mm. x 350mm. x 280mm.; 7.9kg. weight

KEYBOARD

Type: two-tone Selectric-style qwerty with 85 sculpted keys, including 10 function keys and 10-key cursor control/numeric keypad

Features: Intel 8084 microprocessor control including 2K of ROM, 20-key buffer and n-key rollover; legs to provide tilt.

Dimensions: 500mm. x 200mm. x 57mm.; 2.8kg. weight.

PRINTER

Type: 80cps. graphics nine-pin dot-matrix printer, Epson MX-80, with parallel interface

Features: tractor feed; range of print styles; stylish perspex stand is optional extra but recommended as it keeps the cables out of the paper feed

Dimensions: 374mm. x 305mm. x 107mm.; 5.5kg. weight

SOURCE

Manufacturer: IBM, available via dealer network

Contact: IBM United Kingdom Ltd, North Harbour, Baltic House, Portsmouth PO6 3AU

(continued from previous page)

that particular application. The Path structure might then be something like

Path\Multiplan\Accounts\Fred

Fred would avoid all confusion with similar files created by Jim in Sales, whose directory would be found by

Path\Multiplan\Sales\Jim

The program can even include If Exist, to see if a file or directory exists, and Mkdir to create a sub-directory, for example, for a new user, if it does not. But this is not really a multi-user system nor multi-tasking, and would not meet any company's idea of security. The line "Echo Oh dear, someone erased your file" might well come in useful.

Setting up the system obviously involves a lot of messing about with directories, but fortunately DOS 2 provides facilities to do this. For example, Dir|Sort will produce a directory which is sorted into alphabetical order. Dir|Sort>JimFiles will create a file called JimFiles and pipe the sorted listing to it. It can then be displayed on the screen using Type, or printed out. Numerical sorts can be done, and Dir|Sort/25 will sort files into chronological order, that is, by the 25th column which holds the date.

But operating DOS 2 is not all plain sailing, and the Path instruction proved to be a problem in practice; the system will operate happily inside a sub-directory, but will not fetch files from outside it. According to the manual, specifying a Path such as

Path\Multiplan; \Multiplan\Jim; A: \Sales should send DOS to look in the current drive, C, then into the Multiplan subdirectory, then into Jim's Multiplan subdirectory, then to drive A, until it finds the file it is looking for. Whether I am misreading the manual or simply failing to observe the incredibly tortuous syntax I do not know, but I cannot make it work.

When running commercial software the Path command seems to be totally ignored

by DOS 2. Multiplan is one of the few programs that runs happily from the XT hard disc. The Trendtext/2 word processor gave problems by booting from C but then going to drive A for all subsequent files. The program as configured would not even accept C: as a drive identifier, so not even text files could be saved to the hard disc. TK! Solver, reviewed in our August issue, is copy-protected so it has to be run from floppy drive A anyway. However, it refuses to recognise the existence of drive C, no matter how configured. The only way round it is to Assign C to be B — no fun.

Tomorrow's Office is supplied on six floppy discs which makes it a strong candidate for hard disc operation, otherwise you have to change discs the whole time; at its launch the program was demonstrated on the IBM XT. Again, however, the early review sample supplied proved impossible to configure for the XT in our office. Even when it could be instructed to look through C for files known to be on C, the program would hang up while waiting for the user to insert a disc in drive C.

Inserting a floppy into the IBM hard disc is not a pastime to be encouraged, and Sisoft has responded with an improved version of the product to match the XT. However, not every company is likely to meet the challenge quickly, and not every software package will be easy to change. The Bristol Software Factory, producers of Silicon Office, has complained publicly about the situation. In the weekly trade publication *Computing*, August 4, Mike McDonald said he suspected there was a hardware difference in the interface with the machine's operating system which gave difficulties.

So while some programs can be run from one drive some, like Context MBA, require two drives and cannot be run at all. In any event the most likely result is that the poor

user who pays out a large amount of money for permanent ownership of an XT ends up with a single-floppy micro with a built-in 10Mbyte back-up disc.

Obviously this situation is going to change. IBM can currently sell XT's faster than they can make them and a queue is building up outside the sales department. The potential for software sales is immense, and the supply will arise to satisfy that demand. However, it does mean there is little benefit for the ordinary user in being near the head of the queue.

In the long term the XT looks like a winner. The average small-business micro user will find that the ergonomic excellence of the IBM XT, the generous 256K of RAM, and the vast capacity of the hard disc a real boon. Switching from an ordinary eight-bit twin-floppy CP/M machine to the XT is like going from a Metro to a Mercedes. Both get you from A to B but there are differences in style, comfort and convenience, as well as cost.

It is a kind of comfort and convenience that most serious users plan to get used to. With the cost of hard discs dropping significantly at the moment, and with the mass of software and add-on accessories becoming available for the IBM, the XT version looks very much like being the Apple II of the next five years. It is hard to think of a higher compliment than that.

Conclusions

- The IBM XT with monochrome monitor and printer represents a well designed and well integrated system, which has great versatility and no obvious bugs. Ergonomically the system is outstanding.

- The keyboard has an excellent touch, but the placing of four or five keys may create problems for some people.

- Personal Computer DOS 2.0 is larger, more complex and more sophisticated than the previous versions. It is harder to learn, but the facilities offered will repay study. Many DOS 2 facilities are usable on twin-floppy machines as well as on the hard disc version reviewed.

- DOS 2 offers a learning path and an upgrade path into Xenix, the Microsoft version of Unix.

- Basic has been enhanced for the XT, and again the extra facilities are available to non-XT users. The language is not particularly fast or powerful, but contains an enormous number of commands.

- The XT hard disc requires a lot of effort to organise, but after that should prove trouble free in operation. That there is no alternative to backing up onto floppy discs is to be regretted.

- The system as reviewed, with 256K of RAM, graphics printer plus stand, and all cards and cables costs £5,200 plus VAT from IBM Retail Centres. This makes it good but not exceptional value. However, as the price drops over the coming years the XT could well become the standard small-business micro.

Screen display which results from running the logon batch file.

```
YOUR PASSWORD IS BEING CHECKED
PASS, FRIEND
ENTER YOUR COMMAND
A = WORD PROCESSING
B = MULTIPLAN
C = STRIP POKER
```

```
C>CHKDSK
```

```
Volume JACK created Jan 1, 1980 12:04a
```

Screen display from running the check disc utility for hard disc C.

```
10592256 bytes total disk space
 28672 bytes in 3 hidden files
  4096 bytes in 1 directories
708608 bytes in 89 user files
9850880 bytes available on disk
```

```
262144 bytes total memory
237328 bytes free
```

THE VITESSE from Logica is a good-looking relatively simple 8086 micro-computer that comes in a pleasantly designed cream box measuring 34cm. by 46cm. and standing 25cm. high. This main unit holds the processing electronics and a pair of sensible capacity 592K mini-floppy drives. The tiltable screen and its keyboard are packaged as separate modules.

The illuminated main power switch is on the front panel of the processor unit; after switching it on and waiting 10 seconds the screen comes alive with the single prompt

Disk
and a symbol of a rectangle and a backward-pointing arrow. The same symbol is used on the keyboard to identify Carriage Return. With a system disc in drive 1, hitting Carriage Return — or any other key — triggers the CP/M-86 boot sequence. There is no debugging PROM monitor below operating-system level.

In most implementations of the eight-bit progenitor, CP/M-80, the operating system is small enough to fit on the outermost track, track 0, of a floppy disc. The first few bytes of track 0 will be a very simple loader routine supplied by the hardware manufacturer to read in the rest of the system track and make sure it is placed correctly in RAM.

CP/M-86 works in a very similar way, except that the operating system is too large to fit on a single track, and is therefore represented by a file called CP/M.Sys. It still needs a loader on booting up, which is kept on track 0 as in the eight-bit version. ROM initiates the loader, the loader fetches CP/M.Sys and then the system is booted.

LOGICA VITESSE

Chris Bidmead reviews a 16-bit micro from a leading U.K. manufacturer, which is also sold as a dedicated word processor and under the Merlin label by British Telecom.

As far as I know all CP/M-86 implementations work like this, and MS-DOS is similar though its system software is split across several files. The working of the loader is worth mentioning, however, because one of my main criticisms of this machine centres around this point.

The system disc supplied by Logica has only two files on it, CPM.Sys and a file

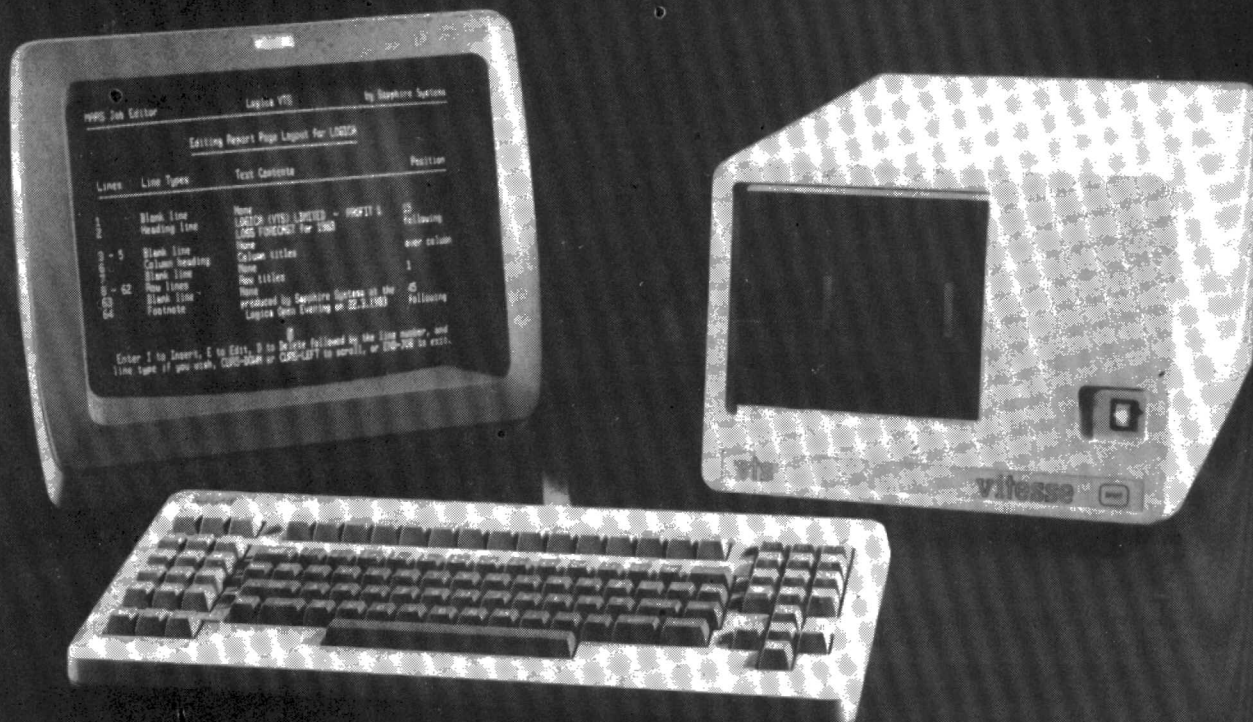
called CPM.H86, which turns out to be a hex version of CPM.Sys. It serves no function, and I am baffled as to why Logica has included it and bothered to document its presence. It would be more helpful to have the rest of the standard CP/M utilities on the same disc, but for some reason they are supplied separately.

The keyboard is uncramped, with

Benchmarks

Running the standard benchmarks on the Logica under Microsoft Basic-86 revision 5.22 revealed a relatively slow machine, considering it uses an Intel 8086 microprocessor like the speedy OEM Orion.

	1	2	3	4	5	6	7	8	Average
Logica VTS	1.8	6.2	13.0	13.5	15.5	28.9	44.9	35.0	19.85
Zenith Z-110	1.5	5.1	10.6	11.0	12.8	24.3	25.5	28.5	14.9
IBM PC	1.2	4.8	11.7	12.2	13.4	23.3	37.4	36.9	17.6
OEM Orion	0.6	2.1	4.8	4.9	5.8	10.5	16.7	13.0	7.3



height adjusters on the underside. The separate key clusters are well spaced out, and two shades of amber are used to differentiate the QWERTY keys from the function and numeric keys. Yet in practice the keyboard is less promising than it looks. Some crucial keys are in odd places: the Control key is on the right-hand side, the Backspace is on the left-hand side, and marked Erase Char, and there is a key called Back Tab where you would expect to find a Backslash.

The top row of the QWERTY keys present a confused appearance, their tops being engraved with three characters rather than the usual two. On some of the keys the additional character is generated by holding down the Special key, but on others the connection between the key top and generated character appears to be arbitrary. The useful feature of Caps Lock is provided to hold alpha characters in upper case without shifting the other keys. It is a common enough feature on computer keyboards, and is usually implemented on a single On-Latch/Off key. On the Vitesse you have to hold down Special and "." to set alpha lock and Special and "," to release it.

Some of the keys carry mnemonics that are valid in the context of CP/M: Clear Cmd sends Control-Z to cancel the command line, Retyp Cmd sends Control-Re, Scrl On/Off sends Control-S. But many others are inscribed with names like Col Retrn, Mode Lock and Erase Word that bear no relation to the operating system or the software provided.

The handily placed array of 12 function keys are unimplemented, beeping at you if you try to use them, and the cursor keys send out control codes that are

echoed on the screen to no very good effect. This last point will come as no surprise to CP/M veterans but, with the IBM PC and so many other new-generation machines offering cursor keys that remain meaningful at operating-system level, would-be customers are going to need some swift sales patter to smooth the rough edges.

The large 15in. amber screen is stable, very easy to read and definitely the best feature of the hardware. It operates in two modes. One is plain and simple with 24 lines by 80 columns while the other offers a message line at the top of the screen, reducing the work area to 22 lines by 80 columns.

The message line carries information about the status of the printer, the position of the cursor and — a useful touch this — translates the current I/O byte into the mnemonics used by Stat and Pip. Thus it keeps you permanently informed about the logical-to-physical I/O linkages.

Ideally the message line would be controlled by dedicated hardware in the monitor, as with the more sophisticated serial terminals like the Cifer range, but on the Vitesse it is a software emulation. Switching it on, using the dedicated SCN Switch key, involves a warm boot of the operating system. If you hit this switch while inside an applications program to see whether the printer is ready, you will be disappointed to find yourself back at the CP/M command line.

The review system arrived with discs for Micromodeller and Mars but no documentation for these programs. Despite repeated promises that the manuals were on their way there was still no sign of them by press day. A more serious disappointment was the absence of Wordsworth, Logica's own word processor, scheduled as the main feature of this review but withdrawn by Logica at the last minute as not yet ready for exposure.

Digital Research's complete documentation for CP/M-86 was provided, along with provisional documentation for Microsoft's MBasic. This language is now effectively unsupported under CP/M due to the internecine strife between the two operating-system vendors.

One improvement CP/M-86 offers over CP/M-80 is the provision of a Help utility that explains how the various CP/M routines work. Logica salesman are going to have to do some more smooth talking to explain why Tod, the standard date and time utility documented within Help, is nowhere to be found on the utilities disc, Copydisk is explained there too:

Copies all information on one disc to another disc, including the CP/M system tracks if they are present on the source disc...

"System tracks" includes the vital loader on track 0 I mentioned earlier and that brings me to my main objection to Logica's approach.

Digital Research recognised very early on that, one valued aspect of the micro-computer, unlike the main-frame, was the user's maximum independence from the manufacturer. In this spirit, CP/M routinely comes with a set of utilities to create new versions of the system on blank discs. One of these is Sysgen, the system-generation utility. Another standard way of creating new system discs is by using Copydisk, directly transferring all the tracks, including track 0, from one disc to another.

As with Tod you will look in vain for Sysgen and Copydisk among the CP/M utilities offered with the Vitesse. Instead there is a utility called Backup, a track-to-track copier written by Logica that expressly omits transferring track 0. Logica is allowing you to create data discs and copy them, but has gone out of its way to make sure you will never be able to generate new system discs.

This is a mainframe marketing strategy designed to tie the user closely to the manufacturer. Readers of this magazine will have picked up the feel of the micro world enough to know that this is not what micros are about. I hope no amount of salesman's rodomontade will persuade them otherwise.

With such an extensive choice of dual-floppy micros available, customers can avoid this problem very simply. If they stay away from the Vitesse on this account Logica can hardly complain. The company might even be grateful — that way it can be absolutely sure nobody is copying its treasured system discs.

Let's hope Logica quickly changes its mind and falls in with the more neighbourly behaviour adopted by the majority of micro manufacturers. It would be a pity if such an amiable machine, with a large friendly screen and a fast, true 16-bit processor were given the cold shoulder on account of an old-fashion, misapplied marketing ploy.

Conclusions

- The Logica Vitesse has been developed from the company's dedicated word-processor the VTS. The development is still in progress, and at the moment the system presents some rather ragged edges.

- The hardware looks good and is pleasant to use. The screen in particular is large, with very legible orange characters.

- The operating system is an incomplete version of CP/M-86, a historic piece of software that has not really seized the opportunities offered by the 16-bit environment. MS-DOS 2 and Concurrent CP/M have been available to OEMs since January. MS-DOS 2 would seem to be a natural choice, as Logica is the U.K. guardian of Microsoft's Xenix.

- Following in the leaden footsteps of Dec, Logica is apron-stringing its customers by withholding the facilities for creating system discs.

Specification

SYSTEM

CPU: 8086 true 16-bit processor

Memory: from 64K to 516K; review model had 256K

Discs: twin 5.25in. 592K drives; literature suggests the intention to offer 1Mbyte drives

Interfaces: Centronics; optional RS-232C

Dimensions: 34cm. x 46cm. x 25cm.

DISPLAY

Type: 15in. orange phosphor

Display: 24 lines x 80 characters with 22 line option; seven-by-nine dot matrix, reverse video, bold, underline

Dimensions: 35cm. x 37cm. x 38cm.

KEYBOARD

Type: detached, international standard full QWERTY pad

Features: Calculator-style numeric keypad with 18 keys; 12 programmable function keys, disabled

Dimensions: 48cm. x 20cm., height adjustable

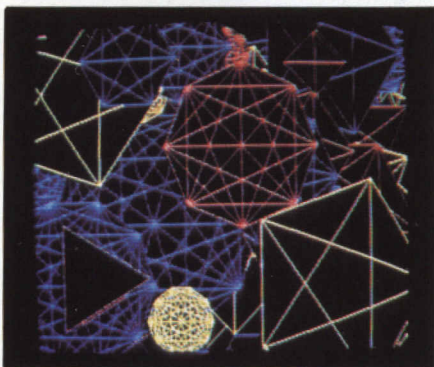
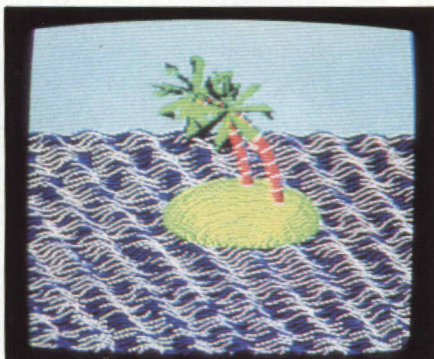
Manufacturer: Logica VTS Ltd, 86 Newman Street, London W1A 4SE. Telephone: 01-637 5171.

Price: £2,490 for 64K system

ELECTRON



Acorn's long awaited Electron is here. It is smaller and cheaper than the BBC Micros, but the machines have a lot in common. Neville Maude thinks it should do well.



A Welcome tape is provided which follows the BBC Micro style. It includes Polygon; Island, where the waves move; and Draw, the horizontals and verticals are fine, diagonals difficult, and curves almost impossible.

Specification

CPU: 6502A running at 2MHz
Memory: two 16K ROM/EPROM chips plus 32K of RAM from four chips
Keyboard: 56 typewriter keys in QWERTY layout
Ports: UHF TV, video, RGB monitor and cassette ports; expansion bus
Features: colour graphics and sound; number keys used as function keys; optional single-key Basic keyword entry; user-definable characters
Notable omissions: BBC Mode 7; no joystick ports
Power supply: separate, 19V 14W
Dimension: 343mm. x 159mm. x 57mm.
Origin: assembled in Malaysia for Acorn Computers, Fulbourn Road, Cherry Hinton, Cambridge CB1 4JN
Price: £199



Technical details

The 6502A processor runs at 2MHz while accessing ROM, but in the Electron at 1MHz from RAM. This is because the RAM is in four 64K by 1-bit chips, for cheapness, so every access needs two operations.

In modes 0,1 and 2 the RAM access the video part of the ULA is interleaved between the 6502A access. For 40µs of 64 the processor is out of action. In mode 3 the processor is running full speed on alternate lines. In modes 4,5 and 6 it runs at 1MHz all the time it accesses RAM. Hence a program taking 10 seconds on the BBC in all modes takes on the Electron about 43secs in modes 0,1, and 2, about 34secs in mode 3, and 20secs in modes 4,5 and 6.

A trick is to draw graphics by shifting the Electron into its faster modes during the drawing period and then back again. The screen display will be somewhat strange during that period but become normal at the end.

The ULA register of mode is in &FE a write-only register, and the operating system uses &0283. So program insert could be something like:

```
500 DEFPROCquick
510 ?&FE07 = &B0
520 ENDPROC
(PROGRAM)
900 DEFPROCslow
910 ?&DE07 = ?&0283
920 ENDPROC
```

Of course, this does not help to speed up programs where the graphic display is used not just drawn, but it helps with those like Persian, in both manuals, where one looks at the results. Times for this are about 34secs on the BBC, 50secs with Procquick on the Electron, 105secs unaided.

THE ELECTRON is small, neat — less than half the size of its ancestral BBC Micros. The finish, including keys, is light cream and mainly plastic which contributes to its light weight.

The mains transformer, 19V 14W, is separate and has an integral three-pin plug, which is rather large, 3.5in. by 2.5in. by 2.65in., excluding prongs. This can cause problems with some switched sockets or double sockets when two plugs are being used. The advantage of having only low voltage reaching the computer is obvious, especially for children, there is also no heating problem in the main casing. The transformer appears to have a thermal overload cut-out — a good idea.

The nominal RAM is 32K, which is not immediately apparent from the instruction books. If one asks the computer how much RAM is spare, with the standard phrase

DIM P%:PRINT HIMEM - P%

the answer is 20,990. It is because the Electron does not support the teletext mode 7. The nearest is mode 6, see table, which needs about 8K as compared with mode 7

which uses 1K. Apart from this omission the modes are the same as for the Model B, not the A — a real achievement in so low-priced a micro. The high-definition modes 0,1, and 2 need 20K as they do with Model B but this is unavoidable, for example, mode 5 permits 16 colours with 160 by 256 pixels. In general the graphics are outstandingly good though slower than the Model B.

The standard question to determine the operating system with these micros is *Help and the Electron replies with 1.0 OS, not the latest 1.2. However, it is versatile with plenty of *FX commands. Indeed, there are a couple which the Model B does not have, namely *FX226 which sets the base number for Func A to P, and *FX227 which does the same for Func Q to /.

There are four sockets on the left of the computer, not the right as shown in the manual, and these are labelled underneath the case, UHF TV, video, RGB, cassette. The video socket is for a monochrome monitor and the DIN socket for the cassette player is for 1,200 baud, not alterable to

300 baud. There is also a multi-pin connector under the body, thoughtfully shielded with plastic in case anyone puts the micro on a metal projection. Presumably this will be used in conjunction with the first add-on for the Electron which is called the Elk. It is a general-purpose module to enable sideways ROMs, printer interface, games paddle sockets and RS-232.

The ULA is a large one, about 30mm. by 30mm. with 68 connections. It controls the colour palette and takes over the CRT controller action of the 6845 in the Model B.

But the Electron has no 6845, so there can be no sideways scrolling as used in games such as Planetoid. Internal timing is also taken over by the ULA, as is sound. This is less complicated than the BBC method. To allow reasonable compatibility between the two micros there are three tone channels and one for noise. However, only one tone channel at a time can be used on the Electron and the envelope is also more simple, most people will find it still complex enough.

If tested for speed using the statutory benchmarks the Electron runs about 30 to 40 percent slower. Arithmetical computations are the slowest but, since the BBC Basic is so fast the Electron is still doing well.

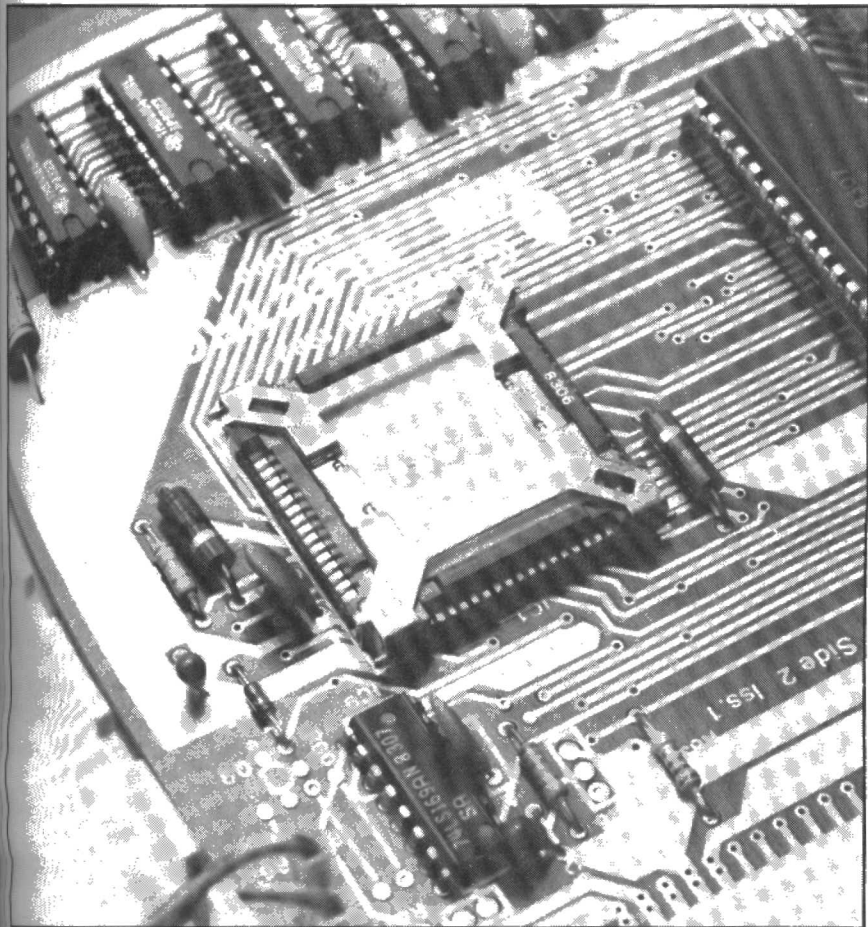
If one tries to load a BBC cassette into the Electron the title page usually comes out as monochrome hash — not always. The main program generally loads but then runs like an arthritic snail, about 2.0 to 4.3 times slower than it should. The Electron does its best, for example, it interprets mode 7 as mode 6 instead of just stopping, and since it cannot implement the double-height BBC command for titles it just prints two identical normal-height lines. The programs on the Electron Welcome tape ran perfectly on the Model B, but at present it is not known if the versions of Snapper, etc., being rewritten for the Electron will be perfectly compatible on the Model B. As a very rough rule, programs for the BBC Micros will not work on the Electron unless altered; programs for the Electron probably will work on the BBC but may not take advantage of all BBC facilities.

The Electron keyboard is a real one, not rubberised plastic, an experienced typist reported that she was perfectly happy with it. The construction is a little cheaper than that of the Model B but is still good. The number of keys has been reduced and both the user-programmable keys and the cursor keys are combined with others. A function key may be used in conjunction with 29 keys to give Basic keywords. For example, Print may be entered in full or as P or Func P, so the Electron has the best of both worlds. There are two omissions, Tab and the shift lock, but those who never had them will presumably not miss them.

In general the Electron keyboard is easier to learn than the BBC and considerable thought has gone into making it simple.

(continued on next page)

Mode	Characters	Pixels	Colour	Memory
0	80 × 32	640 × 256	2	20K
1	40 × 32	320 × 256	4	20K
2	20 × 32	160 × 256	16	20K
3	80 × 25	TEXT	2	16K
4	40 × 32	320 × 256	2	10K
5	20 × 32	160 × 256	4	10K
6	40 × 25	TEXT	2	8K



The ULA is a major reason for the Electron being cheaper than the BBC computers.

ELECTRON

(continued from previous page)

The programmable keys run from 1 to 9 and then 0, as distinct from the BBC 0 to 9 series. The change means that the numeric and f values are the same on the same keys. Only one definition can be put in each programmable key, not three as in old BBC. Small hands will find it easier to reach keys without stretching, a useful point since most Electron users will be young.

The Electron comes with a user guide, 290 pages, in a ring binder. It is smaller than the BBC one, partly because there is less to describe but also because it is written more simply. Apart from not having an index it is a really superb book with better organised information than in the more detailed BBC manual. Those who have trouble with the BBC could try this volume as an alternative, if available separately, since much of the information is similar.

Another book supplied is *Start Programming with the Electron*; again this is excellent, much better than most other books written to help learning to program the BBC computer. One hopes the authors will produce a companion book for the BBC, otherwise this one will help to get started with both.

A Welcome tape is provided which follows the successful pattern with small

improvements from experience. Some programs, such as Patterns, are much the same. Gomuku has come in from the BBC games of strategy cassette, Island is from Acornsoft's graphics book and others are new. A two metre coaxial lead is provided for connection with a television set, production machines will also have a lead for the cassette player.

Many comparisons have been made between the Electron and the BBC micro; unavoidable as the latter is a known machine and the two have so much in common. Nevertheless, in the market place the contest will be between the Electron and micros costing less than £200 — a crowded arena. The Electron should do well as it has many advantages over the present competition. Others will arrive, in particular there are Ataris on the way; the 600XL and 800XL should come in this price range and are said to be compatible with the vast range of existing software. It is not impossible for Acorn to reduce its price should it become necessary. Acorn's decision not to release machines to software houses prior to the launch is interesting. On one hand it gives Acorn about two months lead with its 10 or so cassettes which are the first to be converted, on the other hand software sells computers.

The Electron will go out to dealers and high street chains. Acorn projects sales of 100,000 by Christmas with W H Smith stocking it and then perhaps Boots. The

Electron should carry BBC Basic into many more homes and it is anticipated children will use the BBC at school and the Electron at home. Curry is quoted as saying "The BBC is happy because they see it as support for the language, making it as standard as possible."

Conclusions

● The Electron is an excellent micro for the money. It is rumoured it will sell for £199. It is a little unfair to compare it with the Model B which costs more than twice as much.

● The Electron will sell well at the cheap end of the market place and the first add-on module should be available almost immediately after the launch.

● The Electron is not a replacement for the Model A; the Electron cannot be upgraded to a Model B, as could the A. Even when add-ons are available, which will make it cost higher than a Model B, the result will still be an augmented Electron, not a B.

● Backing will be good; books for the Electron have been written and a users' manual has been announced.

● The Electron has a good keyboard, colour, graphics and Basic plus strong connections in the educational field. It can be recommended as a first computer for which to learn, or as a step up from cheaper types such as the ZX-81.

Up to scratch?

Genuine computer specialists can always come up with the answer. So if you're puzzling about which computer would suit your purpose best, don't worry any more. Just come along to the Leeds Computer Centre. We'll be delighted to advise and demonstrate from our stock which covers the whole international computer scene, with an emphasis on quality and value. This isn't the sum total of what we can do to help you. Perhaps you could use advice on extending your present system. Or would you like to attend our seminars or have employees come along?

The Leeds Computer Centre is equipped with computers and all that goes with them, particularly expertise, to solve your particular problem(s).

All you have to do is ask, preferably in person or initially by telephone.

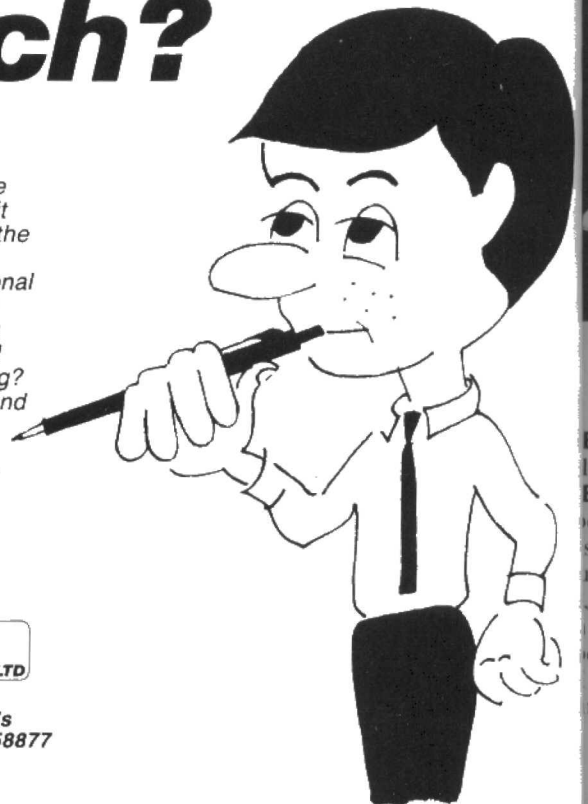
Main dealers for:-

Sharp, Epson, Gemini, Quantum, Nascom & Dragon

THE LEEDS
COMPUTER CENTRE

BTS-PC'S
COMPUTER PRODUCTS LTD

55 Wade Lane,
Merrion Centre, Leeds
LS2 8NJ Tel (0532) 458877



● Circle No. 1

SYMBI
m for th
g develo
patible w
he Appl
BFILES
all DOS,
ware, in
pages to t



BNET is
low sev
BNET
comput
s that S
actor las
where:
are
copiers
running
rent of

Dur

SYMBIOTIC

COMPUTER SYSTEMS LIMITED

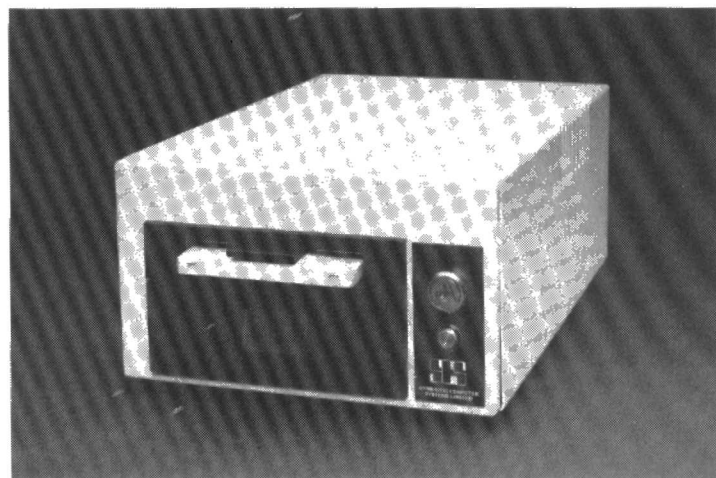


SYMBFILE

5 1/4" WINCHESTER SUBSYSTEM

The SYMBFILE hard disk subsystem is a complete add-on mass storage system for the Apple II, II+, IIe, or III microcomputers and is at present being developed for the SIRIUS, IBM PC and the BBC micro. It is compatible with the majority of hardware products currently available for the Apple, including the 16K Language card and 80-column cards. SYMBFILES are available in sizes from 5-21 megabytes.

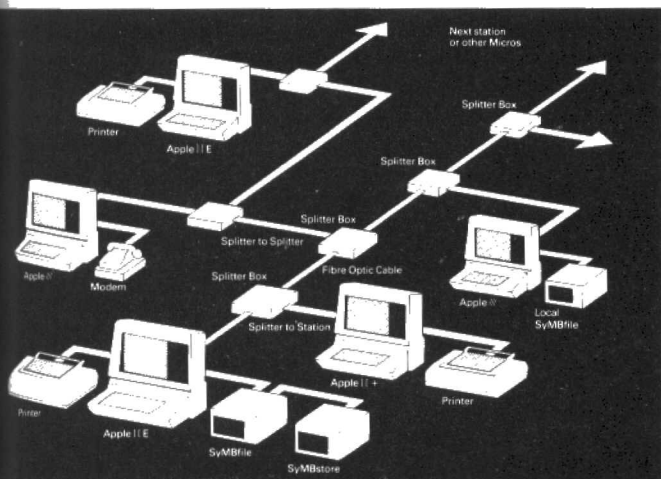
Full DOS, Pascal, and CP/M support allows any standard application software, including database, word processing, and accounting packages to be used.



SYMBSTORE

TAPE STREAMING BACK UP

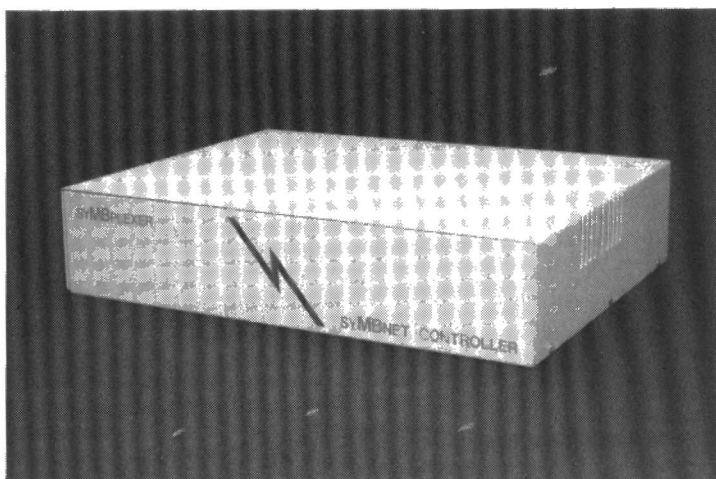
SYMBSTORE is the perfect complement to SYMBFILE giving a high speed, totally reliable back up system. SYMBSTORE will copy the entire contents of a SYMBFILE to a C60 type digital cassette. SYMBSTORE'S unique multiple buffer verification ensures the perfect transfer of data. Software to individually back up volumes and files under present operating systems will soon be available on request.



SYMBNET

LOCAL AREA NETWORK

SYMBNET is a "tree and branch" network system using fibre optic cable to allow several microcomputers to share a common SYMBFILE. SYMBNET is the fastest long range local area network for microcomputers, and can cover a range of 7-9 Kilometres. Fibre optics means that SYMBNET is more cost effective; it uses a high intensity semiconductor laser to transmit data and cables can be laid along the shortest route, whereas other networking systems use flat ribbon or coaxial cable which are sensitive to electrical noise from fluorescent lights, photocopyers, etc. SYMBNET is compatible with DOS, Pascal, SOS, and CP/M running on any microcomputer supported by the SYMBFILE in current operation.



SYMBPLEXER

NETWORK CONTROLLER

The SYMBPLEXER is a network controller which complements SYMBNET. The SYMBPLEXER is connected directly to the SYMBFILE and performs all read-write operations to and from the hard disk. SYMBPLEXER does away with the need for a central machine thus releasing another terminal to run any application you wish, being a dedicated device the read-write operations are performed very efficiently, the support software allows SYMBPLEXER to designate pass codes for each user and to decide which user may access which applications. If you are currently using SYMBNET, accessing your SYMBFILE via a central machine, the addition of a SYMBPLEXER will in no way change the operational capabilities of the network.

For more details of all SYMBIOTIC products contact

SYMBIOTIC
COMPUTER SYSTEMS LIMITED

Duroma House, 32, Elmwood Road, Croydon, Surrey CR9 2TX ☎ 01 683 1137 PBX Telex 893815

● Circle No. 151

A 16-BIT machine built around the advanced 8086 processor with 128K of RAM and a modern business standard keyboard, for £350 — less than the cost of a BBC Model B. An IBM-compatible disc-based system with twin drives with WordStar, Mailmerge and CalcStar thrown in, for £1,200. These two systems from the North London-based company Advance Technology U.K. certainly have remarkable specifications for their price.

Both models, the Advance 86 Model A and the Advance 86 Model B, are scheduled to be launched in September. I had a look at pre-production versions and talked to some of the people behind the systems. What I actually saw was the electronics of the systems without production casing, and pre-production mock-ups of the casing the systems will be delivered in. Advance say September is when it hopes to be actually delivering systems to computer shops.

Externally the Advance looks like a modern business computer. The Model A comes in two units, a system box and a separate detached keyboard on the end of a cable. The Model B comes in a third box containing two disc drives and other goodies. This clips on top of the Model A system unit. So really there is no separate Model B, but rather an expansion unit which converts the Model A into a Model B. Model A users can convert to the disc-based system for £852.

The reason there are two models is to enable the Advance to address two distinct market slots. The model A is aimed at the kind of people who are buying the Commodore 64 and BBC computers.

The disc-based Model B is aimed at the same kind of people as the IBM PC itself, or people who are buying IBM look-alikes, or even eight-bit business systems like the Osborne which have some application software thrown in.

The Advance keyboard would certainly impress most home micro users. It is deliberately very like the IBM PC in layout, but to my mind there are certain improvements. The left Shift key has been moved to a more normal location next to the Z key, the Return key enlarged, the numeric keypad moved slightly to the right to separate it from the main keyboard.

The system box contains the main board with its 8086 processor and 128K of RAM. The box is large and flat and, in the mock up at least, is chocolate coloured. Looking at the electronics which goes in it, it could have been much smaller, but since the idea is to have the Model B expansion unit sit on top it makes sense to have both boxes the same size. When not in use the keyboard can be stored away inside the system unit, so the Advance will not take up too much space on a desktop.

Even the entry level Advance Model A at £347.82, comes with 128K of RAM and this can be expanded on board to 256K. By home micro standards this is enormous. A further 16K of RAM is set aside for the display. The system can put out 25 lines of

Inside the Advance

An IBM look-alike for the price of a BBC — sounds too good to be true. Ian Stobie went along to check it out.



There is no separate Model B, an expansion unit converts the Model A into a Model B.

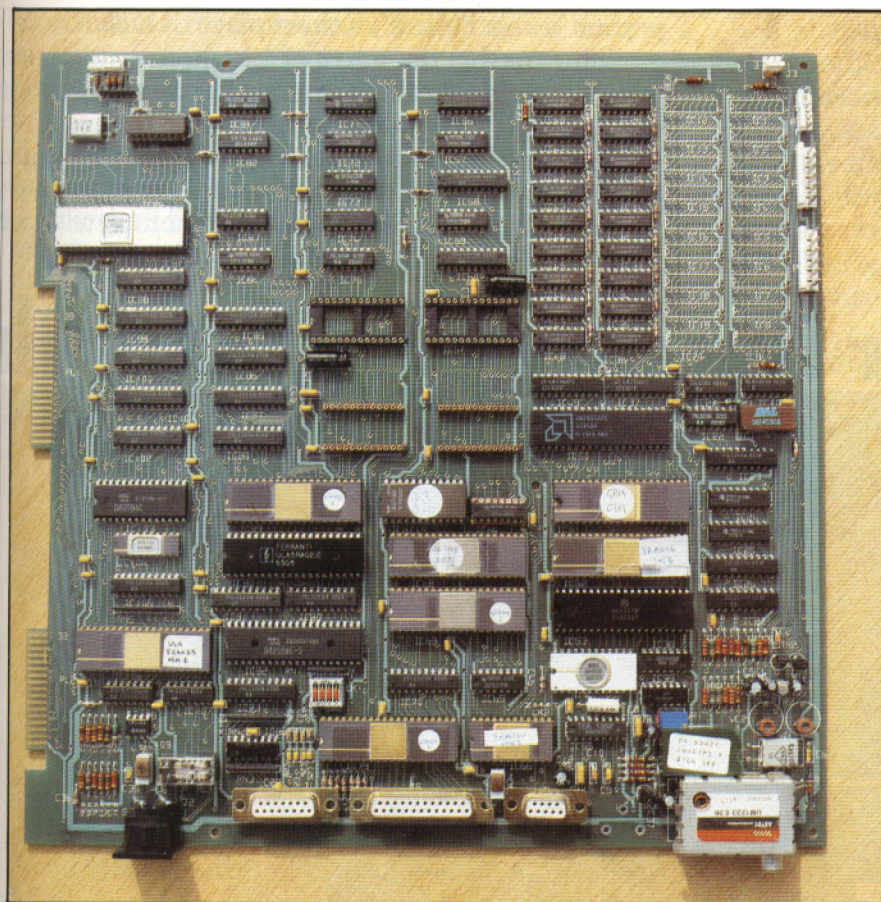
text in either 40- or 80-column widths, or do high-resolution graphics in up to 16 colours. No display device comes with the system but three different outputs are provided, for domestic TV, RGB monitor or composite synch. monitor. A cassette port is provided so programs and data can be stored using an ordinary domestic cassette recorder. The system comes with a joystick port and a Centronics-type parallel-printer port fitted as standard. The printer port in particular is worth having as connecting a printer to systems like the Commodore 64 or Atari can involve appreciable extra cost.

The Model A will run cassette-based commercial software; Advance say they will be marketing a range of titles. In the United States though not in Europe an entry level cassette-based version of the IBM PC has been available, so there is some American software which should run on the 86 Model A. Advance say that with 128K of memory available it is possible for

software suppliers to easily adapt many disc-based packages for distribution on tape; this obviously applies to programs which do not make disc accesses when running but are simply quite large.

The Advance's third unit, the Model B expansion unit, clips on top of the main unit. It is quite simple to fix and no external cables are involved. Once clipped together the two boxes are meant to be treated as one. The expansion unit contains another circuit board and two Shugart 5.25in. floppy drives, providing 640K of disc storage. The Advance's 8086 processor is capable of directly addressing 1Mbyte of memory, and with the Model B expansion unit RAM memory can be expanded up to 768K.

The Advance 86 Model B comes with the MS-DOS operating system, Microsoft GW Basic, an assembler, and three popular Micropro packages — WordStar, Mailmerge, and the CalcStar spreadsheet program. WordStar is the new version 3



The main board has an 8086 processor and 128K of RAM.



The Advance keyboard is deliberately very like the IBM PC in layout.

which has better documentation, horizontal scrolling — and it is in colour.

The man behind the Advance is Jack Dangoor, who is responsible for the overall design of the system and is managing director of Advance Technology U.K. Ltd. "Everybody thought I was absolutely loony when I said I was going to make an IBM look-alike, faster than the IBM with an enormous amount of memory — for the same price as the BBC." Jack Dangoor is no stranger to the consumer end of the electronic market; he has been in electronic watches for a number of years.

Advance Technology is a completely new company set up with private capital with the sole task of selling the Advance computer. It has taken a year and a half to develop and manufacture the system. The Advance core team is very small, consisting

of Jack Dangoor and just three other people. "All manufacturing is contracted out. Advance only sells, nothing else. Once the order is taken it is passed on to the relevant manufacturer who delivers. There are four manufacturers, three are in the U.K. and one is in Japan." Each of these manufacturers is responsible for producing the complete system through to final assembly.

"At least 90 percent of sales will be abroad. Already sole distributors are being appointed on a country-by-country basis. Already there is a distributor for Japan." Jack Dangoor sees the system's IBM compatibility as its key selling point. "You can take a disc for the IBM and just bung it in and it works. As you know there is more software for the IBM now than for anything else. In fact, you can even add a

card for the IBM into the card cage and that will work."

He does not anticipate any copyright problems over his system with IBM. "It has a different processor, it has a different memory architecture, it has a different ROM. It just happens to work the same way. The thing Advance has in ROM is purchased from Microsoft, the same as IBM did. All IBM proprietary work, which is also in ROM, was not copied"

According to Jack Dangoor, the Advance, with its 8086, is a good deal faster than the 8088-based IBM PC. "It is approximately 40 percent faster on average than the IBM. If you run certain games for the IBM, because the Advance is faster, it is more challenging." The Advance has an obvious price advantage. A working Model B system, with disc drives, Micropro software and a cheap monochrome monitor would work out at about £1,250. Jack Dangoor reckons the IBM equivalent would be about £3,400. He is confident that although people might be willing to pay an extra 30 percent for the IBM name, they will not pay that much. He thinks this kind of price difference for both the A and B models is crucial.

Jack Dangoor thinks that most British manufacturers tend to bring out machines and then sell them, for as much as they can get — the wrong strategy for long term success.

"I've been in electronic watches for eight years. If people think that computers are cut throat now they haven't lived. There were two ways with electronic watches. There were the people who made a watch for £100, and then gradually reduced to 95, 90, slowly, slowly, slowly; this to my mind is a typical British computer manufacturer. Then there were the other people: they brought out a watch for £10, and kept it at 10. And in the end they were the ones that won out."

He anticipates eventual competition even at the Advance's price level, but is not worried by the prospect. "The Advance just happens to be the first, and as you know that is everything in this business."

Conclusions

- Both systems seem remarkably good value.

- Software for the cassette-based Advance 86 Model A system may be less abundant than for competing systems, like the Commodore 64 and the BBC micro.

- The 86 Model B is designed to run IBM PC disc-based software, which is very abundant. The software included in the 86 Model B price, especially WordStar and CalcStar, has sold widely and can be recommended.

- The Advance, especially the Model B, stands or falls on the claims made for it as an IBM compatible machine. It should be emphasised that we were unable to use the machine to check how far this compatibility goes.

CP/M's past

For all its popularity, CP/M could hardly be described as elegant. John and Timothy Lee look at what makes it so annoying to use, and find out how many of its faults have been eliminated in the new version, CP/M Plus.

MICROS have now been with us for a decade, during which the CP/M operating system has become the *de facto* standard operating system for eight-bit micros. It was originally written by Gary Kildall, a consultant to Intel, for use on his own Intel development system.

CP/M — the name is said to stand for Control Program for Microcomputers — was then developed and marketed by Digital Research for the Intel 8080 processor, and subsequently for the Zilog Z-80 and Intel 8085-based machines. By 1975 a growing number of microcomputer manufacturers had adapted CP/M to run on their hardware, and a large base of users began to form.

One important feature which made CP/M catch on was the provision of the program ASM. It provided the ability to write machine-code programs using mnemonics, rather than having to hand-code them in hexadecimal. Furthermore, such programs would run on any CP/M machine, making it possible for people to write programs like Microsoft Basic.

With the large CP/M market, programs could be sold at ridiculously low prices compared with the price of software for mainframes. In the early days a revolutionary word-processing program called The Electric Pencil held a position of dominance, though in recent years this spot has been taken by WordStar.

The availability of CP/M and its dependent software led to the widespread use of the Intel 8080 and Zilog Z-80 central processors. The superior speed of the Z-80, which runs at up to 4MHz, together with its much larger instruction set, made it more popular than the original 8080 which can only manage 2MHz. Z-80s are now available running at 6MHz, and even 8MHz. The Intel 8085 which is a code-compatible enhanced version of the 8080 runs at 5MHz and faster. Zilog's forthcoming Z800, a code-compatible enhanced version of the Z-80, will run at up to 25MHz — see Ray Coles' article in the August *Practical Computing*.

The battle for dominance of the 16-bit microcomputer market is still on. CP/M-86 and MS-DOS are strong contenders for computers based on the Intel 8088 and 8086, CPUs while several variations of Unix and CP/M-68 are in contention for the Motorola 68000-based

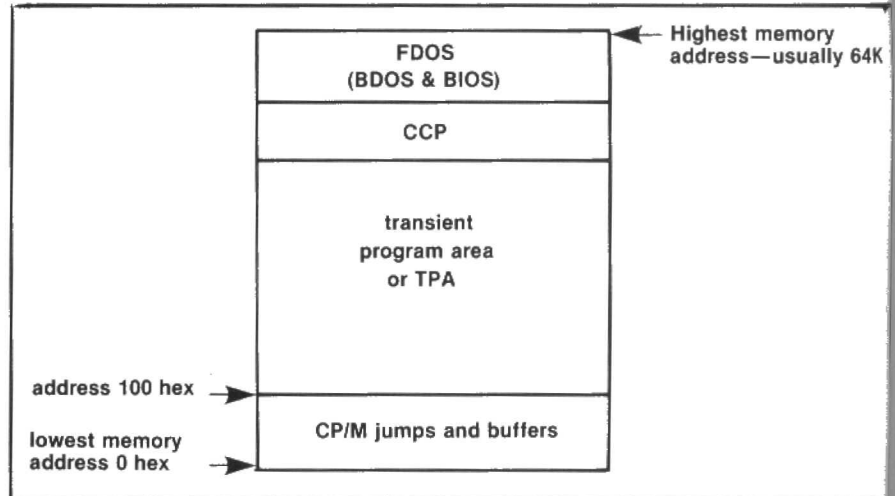


Figure 1. CP/M 2.2 architecture and memory map.

machines. Concurrent CP/M, with time sharing of the CPU between two or more tasks, may prove valuable where the CPU is sufficiently fast and powerful. In this article, CP/M without qualification will refer to the 8080 version of CP/M, release 2.2, now sometimes called CP/M-80. The new CP/M operating system CP/M Plus is the long-awaited CP/M-3.

CP/M does not allow transient programs like Microsoft Basic or WordStar to access more than 64K of memory. At the time CP/M was originally written, this limit appeared astronomically large and unimportant. Now it is the common size. In fact you cannot even get 64K of usable memory since CP/M itself occupies about 7K, leaving only 57K if you are lucky. If your computer has a memory-mapped disc board, or a memory-mapped video board, then even less memory will be available for your program.

Because CP/M is unable to handle more than 64K of memory, there is not enough memory for CP/M to buffer previously used disc sectors. If such buffering were available, sections of data on the disc that are frequently used would be held in buffer memory and would give almost instantaneous program loading. Programs that make extensive use of overlays — WordStar is one — or those word-processors and data-base programs that manipulate large files would run much faster. Some manufacturers have simulated a disc drive using RAM to get

round this deficiency. Such devices are variously known as RAM discs, virtual discs, silicon discs or semidisks, examples being Warpdrive, Semidisk, M-Drive, RAM Disk and Interstellar Drive.

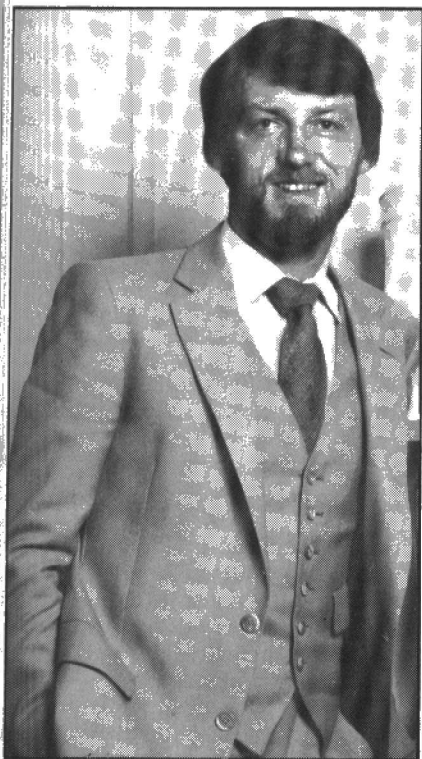
CP/M Plus can be configured in two different ways. The simplest form uses up to 64K of memory, like previous releases of CP/M, and it is called non-banked. However, CP/M Plus also supports multiple banks of memory and this version is called banked.

Configured in banked mode, CP/M puts the TPA user memory in bank 1 and moves most of CP/M to bank 0. Only the top 4K of the users bank of 64K is needed by CP/M, and this 4K must not be bank switching — that is, the top 4K must appear in all banks. This leaves a large TPA of 60K. In the banked version, the CCP is kept permanently in memory in bank 0, so it takes practically no time to return to CP/M command level.

To display the names of the files present on the logged-in disc CP/M uses the Dir intrinsic command. This is fine but the command is slow. Dir works by reading through the file-directory space stored on disc, starting at the beginning and continuing entry by entry until the end. Each time a non-deleted file is found, the name of the file is printed.

Equally important, each time a program opens a file, or looks for a file or a new file extent — that is a new 16-section of a disc file — CP/M has to search sequentially through all of the

and present



Gary Kildall, president and founder of Digital Research, is the primary architect of CP/M.

This results in files being searched for, opened or created much faster. Since files have a directory entry for every 16K of data, this results generally in faster disc access times.

There is little provision for redirection of output. If your CP/M has the IObyte implemented, then you can use Stat to change the console device to any one of three physical devices. But on many copies of CP/M the IObyte is not implemented. You may want to run a program and redirect the output which would normally go to the screen to a disc file, or to a printer. CP/M allows the user to type Control-P for all messages which are sent to the console to be copied to the printer. Unfortunately this does not work when running some proprietary programs like Microsoft Basic.

CP/M does not allow you to send console output to a disc file instead of a terminal. It is only possible to copy output to the printer, and it is not possible to copy console output to a disc file, so it is not possible to create a file containing a sample run of a program.

Similarly CP/M provides only poor facilities for redirection of input. The transient command Submit allows CP/M commands to be read from a file rather than from the keyboard, and Xsub allows command lines to be passed to applications programs. However, these commands only support the passing of command lines. They do not allow single characters to be read from a file rather than typing them from the keyboard. Thus any program that has character commands rather than command lines which have Return at the end of the lines — WordStar for example — cannot be driven using the facilities provided. It should be possible to read input data from a disc file instead of typing it from the keyboard, and redirection facilities of this type exist on all mainframes.

True redirection of input and output is now available on CP/M Plus using the new transient commands Get and Put. Input can now be taken from, or output sent to disc files.

Input from the keyboard is not buffered by CP/M. During a slow disc operation the CPU is not listening to the keyboard, and anything you type during this period will be lost. A good operating system should check periodically to see whether characters have been typed on the keyboard, and store them in a buffer until the program that is running asks for input data. This form of keyboard buffering would prevent characters being lost when

disc access occurs on a word processor.

Early versions of CP/M were designed exclusively for 8in. IBM single-density format discs. The basic units of the IBM format were the track and the sector. Discs had 77 tracks and each track had 26 sectors. Each sector contained 128 bytes of data. CP/M was, and still is, organised around sectors. Files are read or written in 128-byte sectors.

Most disc boards now read or write more than 128 bytes at a time. For example, IBM double-density puts 256 bytes in each sector, and North Star double density puts 512 bytes in each sector. Meanwhile CP/M still works by reading or writing 128-byte chunks of data that CP/M still calls sectors. Thus one, two or four CP/M sectors equal one disc sector. CP/M would run more efficiently if it could be set to work in the appropriate multiples of 128 bytes.

The way CP/M reads discs is wasteful. When a request is made to read a sector from disc, CP/M moves the disc head to the correct track and watches the data passing under it until the required sector is seen. The data is then read into memory. The next file to be read will probably be for the next sector, and it is likely that this sector will be on the same track as the previous sector.

CP/M would run faster if it read and buffered the whole track as soon as the track is first used. Subsequent accesses for this track would then read the data from buffer memory rather than from the disc, and thus would be very quick indeed. Track buffering is not implemented in CP/M although some manufacturers have developed special versions of Bios that buffer a track, for example Turbodos by Software 2000 Inc.

In the banked configuration of CP/M Plus any spare space in bank 0 can be used by CP/M for disc buffering, as can up to another 14 banks of 60K. The total amount of disc buffering available is thus over 850K if sufficient memory is available. Both the banked and the non-banked versions allow a sector count to be set. CP/M always then communicates with the disc system in the multiples of sector-count CP/M sectors. Thus CP/M can be made to work in units of the disc system, or even in tracks.

One of CP/M's annoying and unnecessary features is the need to press control-C to log in a new disc each time you change discs. Cromemco found how to avoid doing this many years ago with its CDOS operating system. If you change

(continued on next page)

directory entries to determine whether the file exists. This is painfully slow and is simply not necessary. Techniques like hashing the directory would reduce the number of disc accesses needed to find a file.

Directory handling has been improved considerably in CP/M Plus as directories are now hashed. When CP/M Plus is asked to create a file, an algorithm calculates into which entry of the directory the file should go. If this entry is empty, the file name is put there, otherwise the algorithm produces another entry to try, and so on until an empty entry is found. If an empty entry cannot be found, then the directory is full.

When CP/M Plus accesses a file, it calculates in which entry in the directory the file name is likely to be, and looks there. The file name will usually be there, but if another file name is found, CP/M Plus tries the next entry where the file name might have been put, etc. If CP/M Plus finds an empty entry before the file name is found then the file does not exist. Thus CP/M Plus usually only looks at one or two entries in the directory to find a file and does not search linearly through all the directory entries as CP/M 2.2 did.

(continued from previous page)

discs and do not press control-C then the first time you try writing to the changed disc, CP/M will stop with a BDOS error.

Copying a whole disc is tedious and slow, using Pip to copy the files, and Sysgen to copy the operating system for the reserved tracks on the disc. It would be more convenient, and much quicker, to have a utility program to copy an entire disc track by track.

With CP/M Plus it is no longer necessary to type Control-C every time a disc is changed. If it tries to write on a disc CP/M Plus detects that the disc has changed and no longer gives BDOS error R/O. Instead it logs the new disc in and does the file write. This improvement should remove one major source of frustration of using CP/M.

CP/M Plus will also search all discs for a program before giving up with an error message. The order in which the discs are searched can be set by the user. Failing to shut the disc door is not fatal.

Even better, an application program can put CP/M Plus in a mode where CP/M Plus never reports an error, but sends a Return code back to the program, indicating that the desired function has not been achieved. Using this facility application programs can be rewritten to put an intelligent error message on the screen, stating the source of the problem and indicating what remedial action should be taken.

CP/M's console command processor, CCP, only looks on the currently logged-in disc for files. It would be more friendly if all discs were checked, starting with the logged-in disc. If the CCP cannot find the file on any disc then a message to this effect should be printed rather than just the file-name and a questionmark.

It is annoying if you type a command line with a spelling mistake and press Return. The CCP does not let you edit the erroneous line to take out the spelling mistake — the whole line has to be retyped instead.

There is a considerable delay when returning to the system from a transient program. For example, when you type System to get out of Microsoft Basic to return to CP/M, there is a considerable delay before the CP/M system prompt A> appears. This is because the transient program may overwrite the CCP, and on returning to CP/M the CCP must be read from disc, and reloaded into the appropriate part of memory — see figure 1.

The transient program ASM, which contributed much to the early success of CP/M, is now very dated. It still works perfectly well, but only accepts the 80 instructions in the 8080 instruction set, thus preventing use of the extra 80 commands in the Z-80 instruction set.

In CP/M Plus the old ASM program has been replaced by a macro assembler, which can also assemble Z-80 code. The IObyte redirection facilities have been

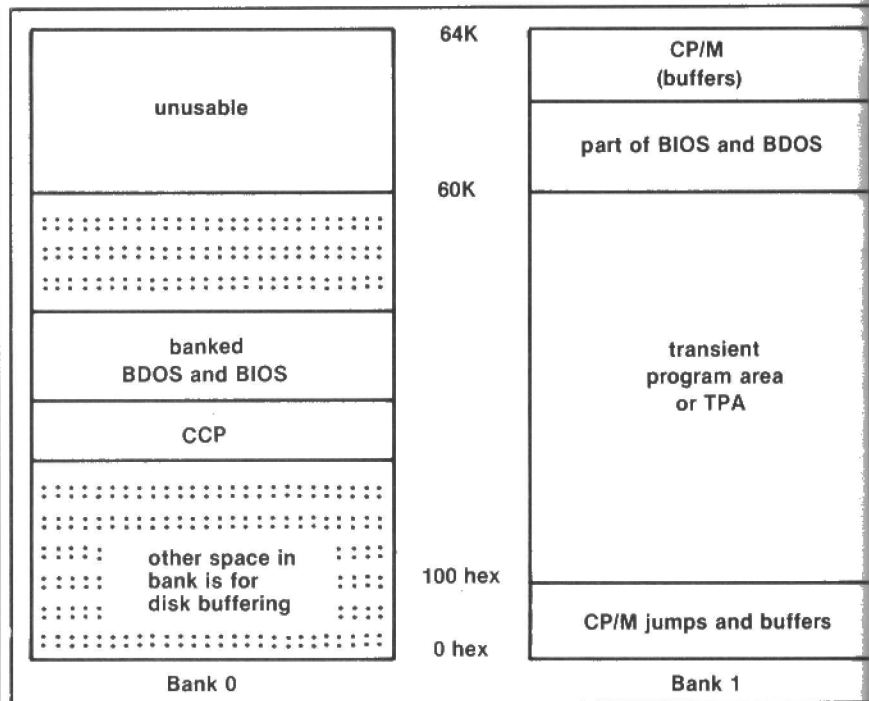


Figure 2. Memory map for banked CP/M Plus.

taken out of Stat and made into a new easy-to-use program. A Help system is also supplied for CP/M. The database used by the Help program can be customised using programs provided.

You can also add help on completely new topics, such as applications programs that you use, or instructions for backing up discs, etc. Pip now has the ability to archive files. Used this way, Pip copies all files that have not previously been archived, and also marks the file as archived. It makes the task of backing up of big hard discs onto floppies slightly more tolerable.

New machines will probably use CP/M Plus rather than CP/M 2.2, as the banked version provides far faster file handling. Installing CP/M Plus will be difficult: Digital Research does not plan to sell CP/M Plus to end-users but dealers will be able to configure and sell versions for their machines.

Those micros that can have lots of memory — for example, all S-100 machines — will benefit greatly from the change from CP/M 2.2 to CP/M Plus. On many micros the discs are the biggest bottleneck, and CP/M Plus dramatically improves disc throughput.

For those who cannot wait until a version of CP/M Plus is available for their machine, a dramatic improvement in computer performance can be achieved by buffering some of the disc in memory. It can be done from CP/M 2.2 with software that is in the public domain and published in *Lifelines* from Lifeboat Associates.

Alternatively you can purchase Warpdrive from Compupro, Semidisk from Semidisk Systems Inc., M-Drive/H from Compupro for S-100 systems, RAM

disc for the Sage, or Pion's Interstel Drive for a wide variety of machines including S-100 systems, IBM, Tandy and Apple. They all provide the extra memory and the software to make it work, usually on a configuration disc.

Effectively these add-ons work by kidding the system into believing that a 256K or 512K memory board is really a disc drive. You can copy files from floppy on to it and use them. If you alter the disc file, then you must copy the file back on to a real floppy disc before switching off.

A cheaper and more subtle approach is to use extra memory as a cache, where only the frequently used disc files or parts of files are buffered. This works with quite a small amount of memory, from 4K upwards. Obviously more memory makes it work better. When the buffer is full, the least recently used part is the first to be discarded. There are two suppliers of this type of system, both British. Ghosia is from Micrology, 4 Deanery Road, Godalming, Surrey GU7 2PQ, and Microcosm Research, 26 Danbury Street, London N1 8JU.

Cifer U.K. is a beta test site for CP/M Plus and is already selling machines with the new operating system. Sirton is also selling CP/M Plus in the U.K. Other dealers will probably follow shortly.

It seems likely that CP/M Plus will add extra life to the eight-bit micros based on the Z-80. At present the 16-bit machines offer the promise of much more power but good 16-bit software is still lacking. Mark Twain said, "Rumours of my death are greatly exaggerated" — and the same is true for the Z-80. In many applications CP/M Plus will increase the throughput giving more delivered power.

Backgammon

BETWEEN THE casing of the cassette and the display on the computer screen Backgammon seems to change its name to Microdeal Pengammon, probably for some inscrutable copyright reason. All the same it is the traditional game of Backgammon.

The program allows the computer to play against you or against itself; alternatively you can use it instead of a board to play with another human, although I cannot see many people wanting to given the Dragon's typical — literally — scintillating display.

The screen displays red and yellow pieces on a green background. Moves are made by typing in the source and destination square numbers, although you can use Microdeal's light pen if you have one. You need to know the rules of Backgammon as neither the screen display nor the packaging help you, and you have no option but to play the doubling game. If the Dragon offers to double the stake and you refuse you have lost.

The game has nine levels of play. You can cheat by changing the machine's level of play during the game or you can get the machine to make your moves for you. The machine seemed to play quite well at its top level, but to be honest I did not find playing Backgammon against a machine sufficiently exciting to provide it with much of a challenge.

Specification

Supplier: Microdeal

Price: £8

Use of graphics: 6/20

Playability: 12/20

Overall rating: 9/20

Cruising on Broadway

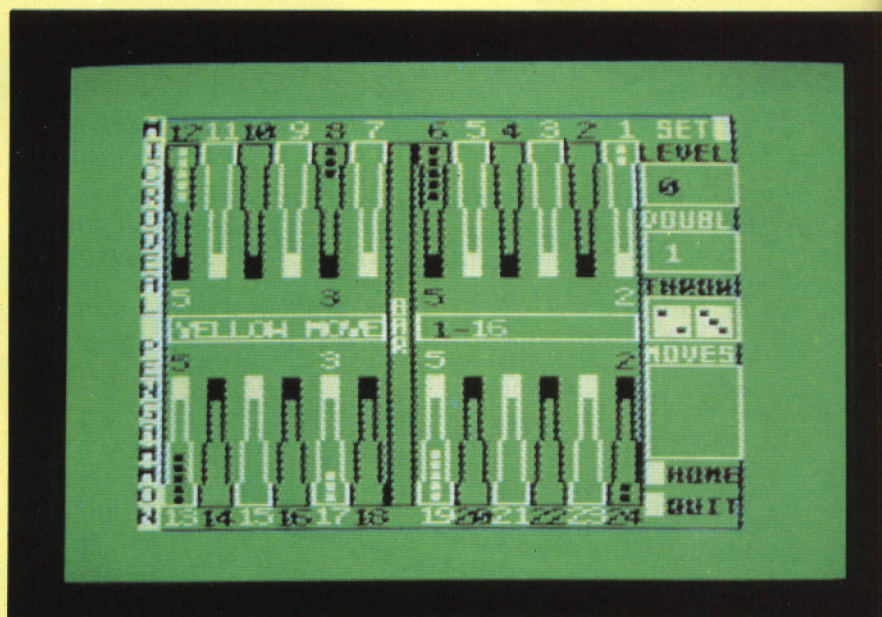
CRUISING ON Broadway is one of the few games to make the transition across from the Spectrum to the Dragon, and is quite a playable game although it is very simple. It has no discernable connection with anything as realistic as Broadway, or for that matter with cruising.

You are a green blob and you are chased by a yellow blob through a simple maze. Success promotes you to higher and progressively more complex mazes until you are eventually eaten. However, your name will live on in the high scoring hall of fame if you can survive sufficiently long.

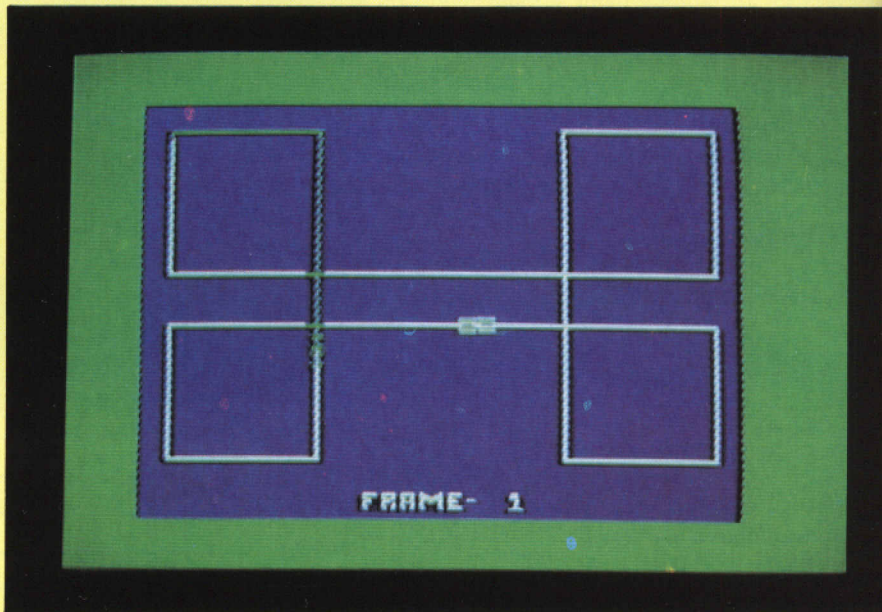
Cruising has simple graphics and sound effects to match, but it is quite compelling and exciting. In fact, the Sunshine logo which comes up as you load the game is one of the best examples I have seen of what can be done with Dragon graphics, although the screens you play on are far less elaborate.

Dragon games

Ian Stobie was not very impressed with the selection of games he tried for this machine.



The Dragon plays Backgammon quite well, but it is not very exciting.



Cruising on Broadway is a game of survival, you can never actually win.

Given the generally poor standard of the 30 or so Dragon games I looked at Cruising must rank as one of the better games available for the machine.

Specification

Supplier: Sunshine
Price: £6.95
Use of graphics: 8/20
Playability: 12/20
Overall rating: 10/20

Gridrunner

SO MUCH goes on in this game it is difficult to describe. It is like a cross between Space Invaders and Centipedes. Your little orange ship is being chased by linked chains of droids across the high energy lattice, the grid. You draw power from the first seven rows of the grid, which you zoom around while firing at the droids. Meanwhile the deadly X/Y zappers try and get you from the side of the grid.

Gridrunner is a top selling game on the Vic-20 and Atari. The Microdeal Dragon version is credited to the same author, Jeff Minter, but it is not as good. It seems slower, the graphics are not as good, and generally it is less exciting.

Much of the problem can be attributed to a less effective use of sound; the game needs lots of noises to generate a sense of excitement. In this version you do not get a noise when you fire. Still, while not initially very compelling Gridrunner is a good game if you persevere.

Specification

Supplier: Salamander
Price: £7.95
Use of graphics: 6/20
Playability: 13/20
Overall rating: 10/20

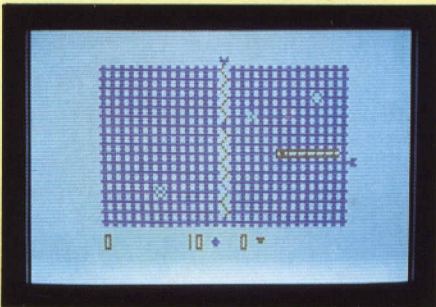
Dragon Trek

THERE ARE several versions of the classic computer game, Star Trek, available for the Dragon and Dragon Trek from Salamander was the best of the three I looked at. It goes beyond the typical text mode display and has reasonable graphics. The game takes place in real time so you cannot take too long over your decisions.

You start by setting the difficulty level, 0-9, and length of game. Your task is to take command of the USS Enterprise and patrol the galaxy, eliminating deadly Klingons to save the Federation. Your ship, armed with three types of phasor and with photon torpedos, is protected by shields. The problem is to correctly use your limited amount of energy. This is used up at an alarming rate whenever



Galactic Ambush is an arcade game.



Gridrunner is not as good as on the Dragon.

your shields are hit by a Klingon, when you fire back, or when you warp or use your impulse jets to move.

This is quite an enjoyable game, but it is still not a patch on the best, truly real-time versions of Star Trek for other machines, for instance, Star Raiders on the Atari.

Specification

Supplier: Salamander
Price: £9.95
Use of graphics: 9/20
Playability: 13/20
Overall rating: 11/20

Galactic Ambush

GALACTIC AMBUSH is a Galaxians-type arcade game. Aliens steadily advance from the top of the screen, occasionally leaving formation to come at you with missiles blazing. You shoot back.

Visually the game is quite good by Dragon standards — the best thing is the three-dimensional moving star field against which the action takes place. But even at the fast speed the game is rather too easy to play. I doubt if it will provide much of a challenge to the average mad gamer for very long.

Specification

Supplier: Microdeal
Price: £8
Use of graphics: 12/20
Playability: 6/20
Overall rating: 9/20

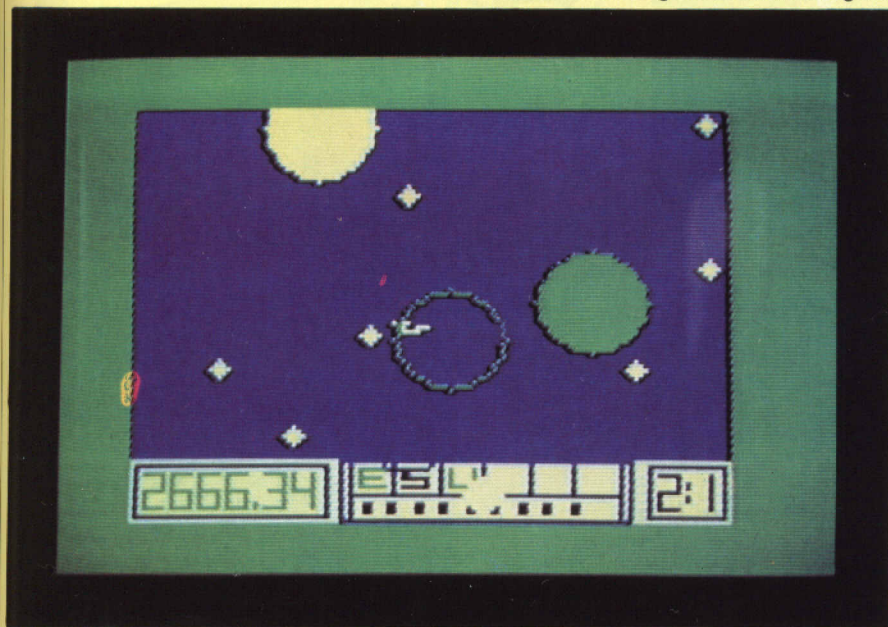
Wormtube

WORMTUBE is a kind of crude Defender, but it is quite enjoyable. You fly your ship through a steadily narrowing tube, scoring more points the further you get along it. Gold nuggets appear in your path, which you have to avoid or shoot apart. You get extra points for gobbling up the fragments of shot-up nuggets.

Up to four people can play Wormtube taking turns with one joystick. The graphics are quite simple but adequate, and the game is made more exciting by the noises which rise in tone as your score increases. This game reminds you that it is not always the most complicated effects that work best. One of the best things about the original arcade Space Invaders, for instance, was the steady insistent noise the ever-more determined invaders made as they came at you.

Specification

Supplier: Hornet
Price: £8
Use of graphics: 6/20
Playability: 13/20
Overall rating: 10/20



Dragon Trek is just one version of Star Trek available for the Dragon.

Oh, so easy WP

Jack Schofield found Atariwriter convenient, easy to use, and relatively cheap.

POWERFUL WORD PROCESSORS are no stranger to the Atari computers: Letter Perfect, Text Wizard and the Atari word processor have been out for three or four years. But the new ROM-based Atariwriter represents a breakthrough in terms of convenience and ease of use, especially for the 16K 400 owner who will be able to use it even with a disc system.

It is also, while virtually as powerful, cheaper than some of its rivals. It costs about a third less than the Atari WP and is half the price of the Letter Perfect ROM, though it is somewhat more expensive than the product it most resembles — Computer Concepts' Wordwise ROM for the BBC micro.

Atariwriter was developed by Atari partly from the Atari WP, in conjunction with Datasoft who produce Text Wizard. It has something in common with both parents, but is most like Text Wizard in its insert mode. Like Wordwise, Atariwriter operates permanently in insert mode so

there is no overwriting. As you type text into the middle of a paragraph existing text is pushed down the screen a word at a time. This creates some odd effects at line endings but is eminently practical.

As with all Atari's main-line programs the documentation and packaging are outstanding. Atariwriter comes with a slim manual which includes tutorial and reference sections, plus a handy quick-reference card. Interestingly the documentation is better than the massive volume and tape provided with the Atari word processor.

To run you just plug it in. The program has two main screens with legible white text on a darkish blue background, darker than the usual Atari screen. First is the menu screen with eight options: Create, Delete, Edit file, Format disc, Index of disc files, Load, Print and Save file. Options are selected by typing the first letter of each word which is shown in inverse type.

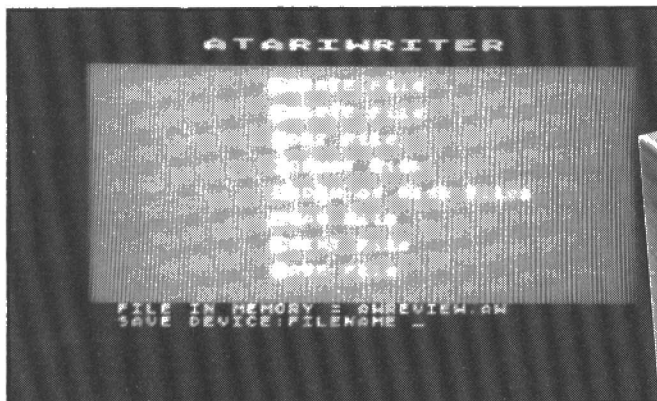
The second screen provides a 21-line by

36-character space for text entry, which is less than the Atari's standard 40-by-24 format. The top line initially contains the default parameters for printing. These can be changed or added to later.

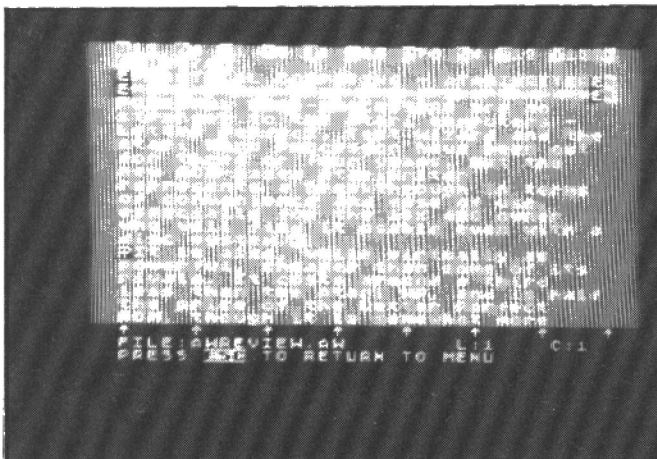
Anyone can use the program without knowing anything about word processing just by typing C and then entering text. The more sophisticated user will learn to use the other functions like block moves and Search and Replace. These are invoked using combinations with Control, Ctrl, and the Atari's three programmable function keys marked Option, Select and Start.

The initial menu screen is self-explanatory and provides some access to Atari DOS functions. Index rapidly alphabetises the list of files, and its scrolling is stopped and started with the space bar. Any function which destroys text or files requires an "Are you sure — Y/N?" confirmation. You can always get back to the menu screen by pressing Esc for Escape.

The text-entry screen provides a



Atariwriters main menu.



Typical Atariwriter screen.



The manual includes tutorial and reference sections.

reasonable range of cursor movements for editing. The basics are handled by the standard Atari full-screen editor with four direction keys, which with Ctrl move you one step at a time in any direction. There is full scrolling up and down, though up-scrolling is a bit jerky. You can go up or down one screenful at a time by Option ↑ and Option ↓.

Four other movements are exactly the same as in Text Wizard. Ctrl A moves the cursor to the start of the line, and Ctrl Z moves it to the end. Select T takes it to the top of the text, Select B to the bottom. There are no word, sentence or paragraph movements.

Deletions are equally simple, by character and by line, using the Delete Back Space key, DBS for short. Select DBS deletes to the end of the file. A 30-line buffer holds the last thing you deleted, so it can be recovered by pressing Start Insert.

The buffer also provides for block moves and block duplication. Each block has to be defined by marking its start and end with a Ctrl X. Again, block deletes require a Y answer to an "Are you sure?" Search and Replace can be individual on the same Y/N basis, or global. The maximum length of a search string is 25 characters — more than generous.

It is always possible to find out how much space remains for text by pressing option F. At this point using a 48K Atari 800, for example, 12,941 bytes — or characters — remain free, so this report will consist of a single continuous file. Atariwriter warns you when there are only 1,500 bytes left. The Atari WP has no limit but you must produce text as a series of saved pages. The memory limits of Letter Perfect and Text Wizard are 36,714 and 30,505 bytes respectively, compared with Atariwriter's initial 26,332 without DOS.

Texts can be merged so it is possible to, say, load a text from disc into the middle of an existing file. Also print files can be chained, a way of handling long articles.

After the text has been entered the next step is formatting for printing. Here Atariwriter is at its weakest because it is limited by the Atari's 40-column screen. One option would be to scroll the screen horizontally to provide a sort of 80-column screen, which is what the Atari word processor does, just like WordStar on the Osborne and Magic Wand on the Apple II. There are 80-column boards for the Atari, which Atariwriter does not mention. Software can also provide 80 columns, though of course the text would not be legible on a TV set.

What Atariwriter does instead is provide a print preview facility, Option P, where the text is set out as it will be printed and the screen forms a window which can be scrolled over it. Thus it is possible to check line and page endings and margins, though it is not very convenient. Many would prefer the Atari word processor preview option also adopted by Wordwise on the BBC, where the text is displayed as it will be printed even though it is illegible.

It does not show how expanded or condensed text will be printed, nor proportional spacing if the printer is capable of it. Also although Atariwriter will print double columns, these are previewed one under the other not side by side.

Being limited to a 36-character screen width for text entry only becomes a real problem when trying to set out tables using the Tab key. If the table is for condensed printing across the maximum 132-character width you really have to construct the table on paper, then type it in afterwards.

The print parameters can be set in half-lines for the bottom margin, top margin, paragraph spacing, line spacing and page length. Widths can be set in characters for left and right margins for two columns, and for paragraph indent. Justification can be on or off. All of these can be varied within a file. Lines can be ranged left, right, or centred. Ctrl O allows decimal codes to be sent to non-Atari printers, such as the



The package is on disc and cassette.

Epson MX-80 used to print this text. Headers and footers can be handled simply with @ providing page numbers. Ctrl E can be used to force the start of a new page.

The final printing out is simply a matter of selecting a printer from the list of four Atari models — select number 3 for a non-Atari printer, and following the screen instructions. You can start and stop at any page and print multiple copies.

There is no Mailmerge capability, as there is with Letter Perfect in conjunction with Data Perfect, but there is a forms capability. If you put an Option Insert character in the text the printer stops at it and waits for an entry — up to 35 characters — from the keyboard. The catch is that the text is not displayed on the screen during printing, and the rest of the line to be filled is probably in the printer buffer. The system is usable if you are careful.

There are a few other facilities missing from Atariwriter — and all the other Atari word processors mentioned. None let you interrupt and resume printing — Atariwriter allows a pause at the end of a page. None will print one file while editing another, or display a second file. None does automatic file back-up. None allows the use of macros to insert key phrases with a single key-stroke, or the use of wild cards in a Search and Replace operation. None provides for soft hyphenation to help with the even spacing of lines. While Atariwriter and its rivals have many qualities, they are not going to replace WordStar and its ilk for the serious writer, though of course it does not aim to.

Conclusions

● Atariwriter is a powerful word processor, well documented and extremely easy to use. It is easier to learn than Atari's Star Raider game, which is supplied on the same kind of ROM cartridge.

● It is suitable for most everyday writing tasks, and coped admirably with the writing of this article. It is not comparable in power to the best CP/M word processors, but provides word processing at a fraction of the price.

● It runs on any Atari micro and can happily be used with discs even on a 16K 400 system.

● At around £65 it is good value and can be recommended.

	Atariwriter	Atari WP	Text Wizard	Letter Perfect
Menu driven	Yes, one	Yes, many	No	Yes
Text insert mode	Yes	No	Yes	No
Global search/replace	Yes	Yes	No	No
Horizontal scrolling	Only in preview	Yes	No	No
Print preview	Via window	Yes	No	No
Double-column printing	Yes	Yes	Yes	No
Prints half-lines	Yes	No	Yes	No
Can edit programs	Yes	No	Yes	No
Mail merge	No	No	Extra	Extra
Pause for text entry from keyboard	Yes	No	No	No
Disc interface	Atari	Atari	Atari	LJK
Medium	ROM	Disc	Disc	Disc or ROM
Producer	Atari/ Datasoft	Atari	Datasoft	LJK
Price	£65	£99.99	£68.95	£109.95, disc £149.95, ROM

This table is not a comprehensive comparison of products, but shows how Atariwriter combines most of the best features of its rivals.

The state of the graphics art

The computer has had a revolutionary effect on games, business and the film industry, to mention but a few — but it is still early days.

IT IS ONLY a few years ago, 1976 to be precise, that computer graphics meant — for most people — Snoopy printed out in a pattern of Xs. To watch the old Teletype print out a naked lady was really awesome.

The following year the Pet micro brought block graphics to thousands. Pictures could be drawn on a screen, not in letters but in little squares, lines and other useful shapes. Then the revolutionary Apple II, in spite of a curious lack of lower case letters, brought user-definable shapes and bit-addressable graphics to anyone with a lot of effort and £2,000 to spare — in colour, too.

Today for under £100 it is possible to buy a colour micro that is capable of drawing, within limits, almost anything the programmer sets his mind to. Screens, whether TV sets or monitors, are the universal method of displaying output. The teleprinter is worth its weight in scrap metal as more and more dot-matrix printers gain the ability to print complicated graphics which may be dumped straight from the screen.

These developments have had a powerful effect on the computer games business. The old Pet version of Star Trek, played in black and white with two axes and a handful of alphabet, pales into insignificance when compared to today's high-speed all-colour all-action arcade games.

The effect on business computing has been quieter but no less revolutionary. It is still possible to use a computer to spew out columns and columns of incomprehensible figures. However, many project managers have found that a graphic flow chart has more impact. Salesmen and accountants have found that an appropriate graph, bar chart or exploded pie diagram can make the point quicker and more forcefully. Sales are going up, or down; the company's share of the market looks like this.

Graphic representations of figures are no longer confined to slide shows and audio-visual displays. They are commonplace in company reports and the financial pages of the best newspapers. This is partly because with inflation and the chaos of the international currency markets, few people have any grasp of what figures mean any more. The important thing is the trend. Graphics provide wonderful opportunities

for massaging figures into attractive shapes — there are lies, damn lies and graphics.

Cynicism aside, business graphics can have a real value, and numerous software packages exist to provide any user with the facilities to produce them simply. VisiCalc, for example, links to VisiPlot and VisiTrend. The current fashion is for spreadsheet and calculation programs to include graphics as part of the package. Lotus 1-2-3, Context MBA and TK! Solver are examples. Graphics are an essential part of integrated operating systems such as Apple's Lisa. Companies like Hewlett Packard, Rikadenki and many more have developed the plotters which will draw suitable graphics with multicoloured precision.

Microcomputer graphics has come a long way in the last five years. Nonetheless there is still a long way to go. Displaying graphics remains a problem, in that TVs and most monitors cannot cope with real high-resolution graphics of 1,024 by 1,024 picture-points or pixels. Indeed, many micros are used with TV sets that are incapable of displaying even the limited resolution they are capable of generating.

However, even higher display capabilities should shortly become common on personal computers, thanks to the remarkable new NEC μ PD 7220 graphics chip. This was runner-up in a recent American hardware-innovation-of-the-year competition, where the winner was the IBM Personal Computer. Two of these chips are used in NEC's Advanced Personal Computer to provide graphics resolution of 1,024 by 1,024 pixels, though the screen only provides a 640 by 475 pixel window onto this. Nonetheless, the display still requires 384K of dedicated video RAM. Even in these days of decreasing RAM prices, this is far beyond the reach of the home micro user and hard for many businesses to justify.

A real high-resolution colour display needs about a megabyte of RAM, but with 256K-bit RAM chips on the way, even this will become widely available in time. Another factor limiting the advance of computer graphics is the lack of standardisation between machines. A comparison of screen displays on small micros reveals every standard from excellent, as on the Acorn BBC Micro, to

the truly appalling, such as the Dragon. Business micros ought to be more homogenous, but in fact are not; even half-a-dozen IBM PC look-alikes turn out to offer different screen resolutions. Such variations limit software portability because almost every graphics routine has to be rewritten to suit each micro.

The solution, suggested at the American ACM Siggraph conference in 1977, is for a core graphics system. The idea is similar to the idea behind CP/M, where all the machine-dependent parts of the operating system are collected together in the BIOS, Basic input/output system. This, in theory at least, is the only part that needs to be rewritten for CP/M to run on different microcomputers. CP/M of course treats the screen display like a primitive Teletype terminal, which is why it currently does not lead to too many problems with the graphics display.

The Siggraph idea was to gather the machine-dependent graphics routines into a similar framework called the CGS or core graphics system. Applications programs would then present a common face to the CGS, which would translate their requirements to suit the particular machine in use. Thus programs could be more standardised and software portability greatly increased. Digital Research will shortly implement the idea in its GKS graphics kernel — addition to CP/M.

Such approaches represent a small step on the right road, but the computer graphics business is by no means settled yet. Systems like the Xerox Star, ICL Perq and Apple Lisa are still making pioneering advances in business graphics and CAD/CAM — computer-aided design and manufacturing applications. The moving graphics of arcade games such as Atari's Pole Position remain a terrible indictment of the graphics capabilities of most home micros. Beyond these there are computer graphics systems which require vast amounts of mainframe processing power, whether for films like Walt Disney's *Tron* or for more serious applications such as modelling or flight simulation for pilot training.

Computer graphics may have come a long way in a mere five years, but the changes over the next five should be equally dramatic.

□

Taking your TV for granted

Do you plug your micro into the first cheap display screen available? A little more information could help you get better results, says Chris Naylor.

MOST OF US can hardly remember a time when we did not have television to watch at home. Now computer users surely look back in vague awe to those times when output did not go to a screen.

The TV screen has become so much a part of our lives that the way it works seems almost to be beneath our attention.

In some ways the attitude is justified. After all, we just want to switch on and watch. Given a small, cheap micro we now want a cheap display screen. The domestic TV is the obvious first choice, but will it give good results? Perhaps a different model would work a bit better, or maybe a special-purpose monitor. The arrival of personal computers has been so sudden that your usual source of such information, the man in the local shop, will not know the answers either.

The cathode-ray tube or CRT is at the

heart of every screen. The tube itself is made of glass, and is evacuated. At the thin end there is a heated element called the cathode, which is negatively charged and emits large numbers of electrons. Left to their own devices these electrons float off in all directions. However, further up the tube is a positively-charged series of plates, which make up the anode. Because electrons are negatively charged they are drawn towards it.

By the time the electrons get to the anode they are going so fast that they cannot stop, so they go charging on and hit the wide end of the tube. The wide end of the tube is coated with a phosphorescent material which glows when the electrons hit it, so when the tube is switched on the wide end glows with a sort of blurred light.

But not everyone wants a tube which

glows evenly all the time, so there is another electrode, called the grid, in front of the cathode at the thin end of the tube. When this grid is lying idle it has no effect but if you place a negative charge on the grid the electrons will not get a sight of the positively charged anodes further up the tube and so will have no incentive to go there. The result is that as the negative charge on the grid is increased, the flow of electrons diminishes. The glow from the bombardment of electrons on the wide end of the tube diminishes too, and eventually ceases altogether when the charge on the grid is large enough.

A glowing tube whose brightness can be varied would be fine to light the room, but not much good as a display medium. What is needed is a little more control, so a ring of electromagnets and plates is placed around the neck of the tube to form an electromagnetic lens. Typically there are three of them and they bring the electron stream to a sharp focus as it approaches the centre of the phosphorescent screen. Electrostatic lenses are also possible. The focus control on the CRT adjusts the electron lens, and the brightness control adjusts the grid voltage.

To make the dot do something a little more interesting there are four plates arranged in pairs around the neck of the tube. If one plate is charged negatively and the opposite plate positively the electron beam deflects towards the positive plate. As there are two sets of plates at right angles to each other the glowing dot can be moved to any point on the screen.

There is just one more thing to worry about and that is the phosphorescent coating on the screen. A phosphor can glow even after electrons have stopped bombarding it. Some phosphors glow longer than others, though in general the glow does not really last long at all. There are three well known phosphors which can be used to coat the tube: fluoresein, which glows yellow-green; quinine sulphate, which glows blue; and chlorophyll, which glows red. By using any one of them or a mixture you can make the moving dot glow in just about any colour you want.

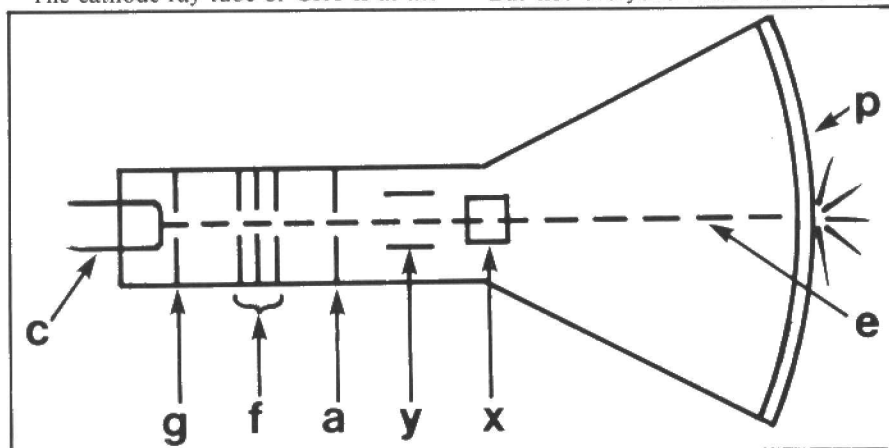


Figure 1. The cathode-ray tube.

- c — cathode. Negatively charged and emits an electron stream.
 - g — grid. By increasing the charge on the grid the electron stream can be reduced, so it acts as the brightness control.
 - f — focusing anodes forming an electrostatic lens to focus the electron beam to a fine point on the screen; the focus control, if there is one, alters the charge on these anodes.
 - a — accelerating anode. Positively charged to draw the beam at high speed towards the screen.
 - x — X plates. Carry an electrostatic charge which deflects the beam from side to side.
 - y — Y plates. Carry a similar charge but work up and down, at right angles to the X plates.
 - e — electron stream.
 - p — phosphorescent coating. Glows when struck by electrons; the colour of the glow depends on the type of phosphor used.
- Some CRTs use electromagnetic fields rather than electrostatic deflection plates to control the electron beam, but the basic principles are the same.

Figure 2.

Using on plates on up and d Using the scope, the X as well waveform interval as the ti cycle of waveform

With why not cheap. I screen, from a couple on your inputs a on the vector anywhere send the

The s does no write a and the it keeps does so deceive stable,

The redraw of the around compu you do is better might t second second fashion

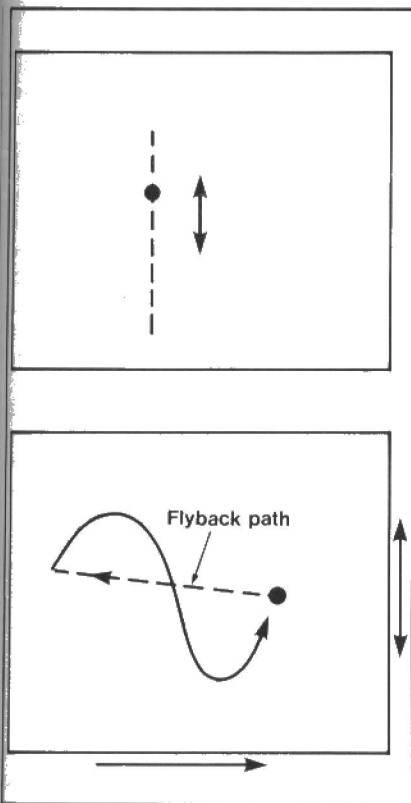


Figure 2. The oscilloscope screen.

Using only one input to control the Y plates only allows the dot to be driven up and down, which is not very useful. Using the automatic timebase on the scope, the dot can be steadily driven in X as well as Y to show the entire waveform plotted against time. If the interval between flybacks is the same as the time taken for one complete cycle of Y input then a complete waveform can be viewed.

With a device like the CRT available, why not stay with it? It's simple and fairly cheap. It can draw its dot anywhere on the screen, so surely it can handle the output from a computer. You just have to put a couple of digital-to-analogue converters on your computer to provide the X and Y inputs and you can drive the dot anywhere on the screen you like. This is what a vector scope does: it can draw anything, anywhere just as fast as the computer can send the X,Y data.

The snag is that the glow from the dot does not last very long, so you have to write a program to drive the CRT in X,Y and then put that program in a loop so that it keeps on driving the CRT in X,Y. If it does so fast enough, the eye will be deceived into thinking that it is viewing a stable, permanent image.

The speed at which the image must be redrawn depends on the flicker-fusion rate of the human eye, which is typically around 12 cycles per second. For a computer that is not very fast at all, but you do have to keep on doing it. Faster still is better, and to be on the safe side you might try doubling the rate to 25 cycles per second. Certainly, below 12 cycles per second the image will start to flicker in a fashion beloved of stroboscopic lighting

Screen test in Apple Basic.

```
10 REM :PRACTICAL COMPUTING SCREEN TEST
11 W = 1:CH = 40:CV = 20: REM :CH IS NO. OF CHARACTERS HORIZONTAL, CV IS C
    HARACTERS VERTICALLY
15 REM :CHECK FULL OUTPUT
16 HOME :A$ = "CHECK FULL OUTPUT": GOSUB 1000
20 TEXT : HOME : INVERSE
30 FOR I = W TO CH * CV
40 PRINT SPC( W );
50 NEXT
55 GOSUB 1000
60 REM :CHECK SCREEN SHAPE
65 HOME :A$ = "CHECK SCREEN SHAPE ": GOSUB 1000
70 HGR
80 U = 159:H = 279: REM :H IS HORIZONTAL AND U IS VERTICAL
85 R = U / 4
90 HCOLOR= 3
100 HPLLOT 0,0 TO H,0 TO H,U TO 0,U TO 0,0
101 HPLLOT H / 2,U / 2: FOR I = 0 TO 6.3 STEP .04: HPLLOT H / 2 + R * COS
    (I),U / 2 + R * SIN (I): NEXT : REM :THIS DRAWS A CIRCLE IN THE CENTR
    E OF THE SCREEN RADIUS R
105 GOSUB 1000
110 REM :TEST LOW FREQUENCY RESPONSE
115 HOME :A$ = "CHECK LOW FREQUENCY RESPONSE": GOSUB 1000
120 TEXT : HOME
130 NORMAL : PRINT SPC( 20 );
140 FOR I = W TO CV / 2
150 INVERSE
160 PRINT SPC( CH );
170 NORMAL
180 PRINT SPC( CH );
190 NEXT : PRINT
195 GOSUB 1000
200 REM :TEST HIGH FREQUENCY RESPONSE
205 HOME :A$ = "CHECK HIGH FREQUENCY RESPONSE": GOSUB 1000
210 HGR
220 FOR I = W TO H - W STEP 2
240 HCOLOR= 3
250 HPLLOT I,0 TO I,U
260 HCOLOR= W
270 HPLLOT I + W,0 TO I + W,U
280 NEXT
285 GOSUB 1000
290 GOTO 15
1000 UTAB 22: PRINT A$: PRINT "PRESS ANY KEY TO CONTINUE ": GET A$: RETURN
```

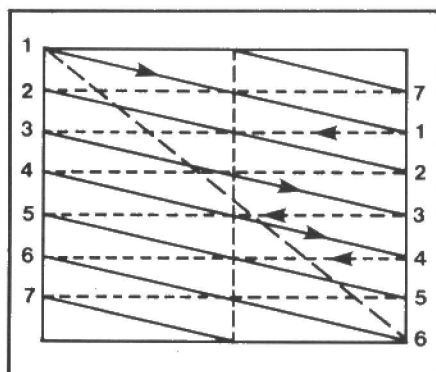


Figure 3. Interlaced scanning.

An interlaced scanning pattern for a seven-line system. Solid lines are drawn on the screen and scanned from left to right. Dotted lines are not drawn on the screen and represent flyback paths. In the first scan the odd-numbered lines are drawn; in the second scan the even-numbered lines are drawn. The diagonal flyback from the end of line 6 to the beginning of line 1 and the vertical flyback from halfway through line 7 are field flybacks. The horizontal flybacks are line flybacks. In the 625-line U.K. system 312.5 are covered in the first 1/50th of a second and the remaining 312.5 covered in the next 1/50th of a second.

but not from side to side. Movement in the X direction is achieved automatically by the scope itself, and is usually called the timebase.

Suppose that you wanted to look at an image of a 1,000Hz audio tone. You place this signal on the Y input of the scope and the dot moves up and down 1,000 times per second, which is much too fast to detect by eye. However, if you slowly move the dot in the X direction at the same time the dot marks out the curve of the 1,000th wave being input. If the dot moves right across the screen 1,000 times each second, you would see one complete cycle of the input tone on the screen.

From a computer person's point of view, the oscilloscope illustrates two important points: the automatic generation of an X input to sweep across the screen, and a very rapid flyback to the

(continued on next page)

(continued from previous page)

starting point so that the process can begin all over again. Using these methods it becomes possible to draw a two-dimensional picture on the screen using only one input rather than having to drive the screen in both X and Y.

At this point we come to television. The devices I have talked about so far can only draw things on the screen that can be represented as line drawings. But remember the grid and the way that it controls the brightness of the dot. If one input is fastened to the grid, the brightness of the dot can be varied at will. The dot is moved rapidly in both the X and Y directions so that it covers the entire screen in a very short time, a technique known as raster scanning. British TVs draw 625 lines 25 times per second so that a whole screen picture is drawn at twice the flicker-fusion rate. To make the picture appear even more stable the scan is interlaced: it is drawn in two passes, each lasting 1/50th of a second. In the first pass all of the odd-numbered lines are drawn, and in the second pass all of the even-numbered lines are drawn — see figure 3.

To synchronise an incoming TV signal with the TV set the signal includes some control pulses which set up the correct line synchronisation, or horizontal scanning and flyback, and frame synchronisation, or vertical scanning and flyback. In between these control pulses, in periods lasting just 1/15,625th of a second, comes the picture signal proper in the form of a burst of activity during which a varying voltage controls the grid on the CRT to vary the brightness of the spot at any given instant — see figure 4.

It is this video input which drives the set. The voltage level controls the brightness of the spot: a high voltage extinguishes the spot to give black on the screen, and a low voltage makes it bright, giving white. In

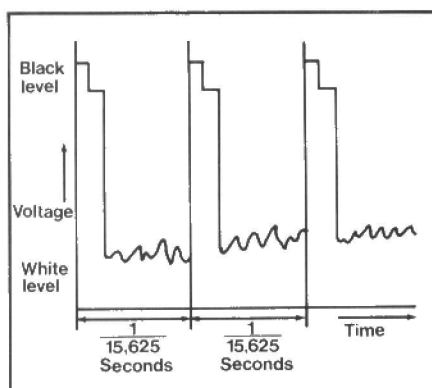


Figure 4. Video input to a monochrome TV. One line is drawn every 1/15,625Hz of a second. The first part of the video signal is a high-voltage line-synchronisation pulse, followed by a slightly lower voltage corresponding to a black tone, during which the flyback occurs. The signal proper then follows. A synchronisation pulse initiates the next line flyback and another line is drawn. A similar method is used at the end of each frame scan to initiate a frame flyback.

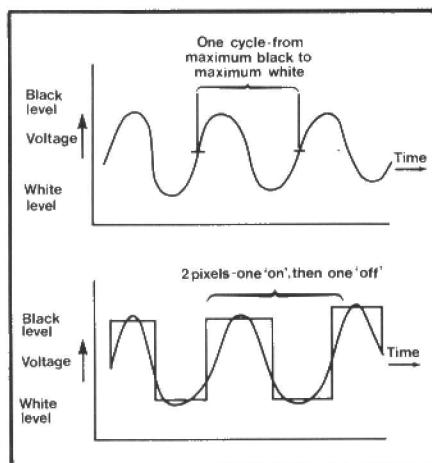


Figure 5. Bandwidth requirements.

The vertical lines on the broadcast test cards are actually sine waves. To display 336 black-and-white vertical bars in each line scan of 1/15,625Hz of a second requires a bandwidth of $336 \times 15,625$ Hz, or 5.25 MHz. If your computer tried to send 336 black and white dots to each line it would almost certainly generate a square wave waveform. It could be approximated by a single sine wave of 5.25MHz, but not very well. The sharp edges of the square wave need higher frequencies to fill in the corners accurately. If the square wave has a frequency f , it will also generate sine waves at $2f$, $4f$ and so on. It will expect, in this example, a bandwidth of 10.5MHz if you try to generate a screen picture 672 pixels wide.

the early days of TV the system worked the other way around.

In a perfect world every TV set would display a perfect picture, but in reality there are all sorts of disturbances in the incoming signal. Such unwanted information in the signal is called noise. The most common type of noise consists of high-voltage spikes which should not be there. With the old system they showed as a white spot, and early TV sets were plagued with a snowy appearance due to noise. The black spots produced by the more modern system are much less noticeable.

In normal TV reception the video signal is used to modulate an ultra-high-frequency carrier wave which is then broadcast. This UHF signal is picked up by an aerial and demodulated in the set to recover the original video signal.

If you want to drive an ordinary TV set from a computer you need a UHF modulator inside the computer to make the signal look like the normal broadcast TV signal. Since the computer's output does not have to be broadcast, it is clearly a waste of time to modulate the signal and then demodulate it again. Specialised monitors and some TV sets have a video input which will accept an unmodulated signal.

A computer drives the screen via a specialised collection of chips. The

(continued on page 96)

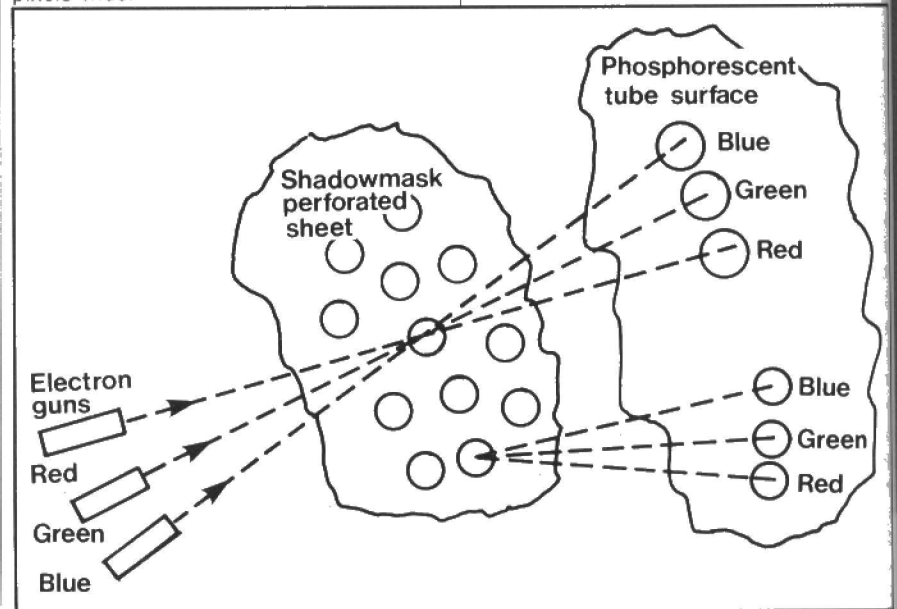


Figure 6. The shadow mask tube.

The shadow mask colour tube has three guns all angled inwards. All three electron beams meet at a single point so that they can be focused and deflected as one by arranging the controlling electromagnetic fields to act at this point also. Directly before the phosphorescent tube surface is the shadow mask, a sheet with thousands of tiny perforations. All three beams pass through each hole in the mask, and so for each hole there are three dots of phosphor on the surface of the tube — one red, one green, and one blue — arranged exactly so that the beam from each gun hits exactly the dot that belongs to it. By controlling the output from each gun an impression of any colour can be formed at any point on the surface of the tube. Unlike the monochrome tube the beam cannot really be directed to any point on the surface, but only to those points corresponding to holes in the mask. On black matrix screens each individual dot of phosphor is surrounded by an opaque black ring which improves the colour definition. The main exception to this general arrangement is the Trinitron tube which uses a system of vertical slots in the mask rather than individual holes. The Trinitron method is said to give a brighter picture.

(continued from page 96)

standard signal formats for driving TV sets are so well known that there really is no reason for the video generator to produce a bad signal. Yet you will not always get a perfect picture: for instance, if the line and frame synchronisation is wrong you will get a complete mess. The problem is too clear-cut to be likely to arise, but you can still find your picture is less than perfect without being a complete disaster.

In a monochrome set bandwidth and resolution dictate how much detail you can see. The Apple II in Hires mode will draw 279 dots on one line horizontally, say a succession of bright pixels and dark pixels — 139 bright pixels in all. This pattern is roughly equivalent to a wave going up and down 139 times. It has to do so within the $1/15,625\text{Hz}$ of a second it takes the line to cross the screen, so the frequency of the signal is 2.17MHz . Any set should be able to cope with a bandwidth like this.

Broadcast test cards have a series of vertical gratings from which you can judge the bandwidth of the set. The highest frequency, corresponding to the finest grating. If your set can resolve the lines on the finest grating then its bandwidth is 5.25MHz at least.

There is an important difference between TV test cards and a row of computer generated dots. The gratings are not real, vertical bars but are actually sine waves — they do not start and stop with a sharp edge. The computer's pixels do have sharp edges, and this raises the bandwidth requirements. The Apple Hires dot is a square wave which requires a whole series of higher-frequency sine waves to represent it accurately.

To fill in the corners with a frequency twice that of the basic signal raises the bandwidth to 4.34MHz for good graphics — see figure 5. Doubling the frequency again brings the bandwidth to 8.69MHz , and at this point, the bandwidth requirements start to exceed the capabilities of most domestic TV sets. The broadcast frequency allocations allow only 8MHz per channel, so that is all a TV set normally has to cope with. Purpose-made monitors can have bandwidths of, say, 24MHz , which is easily enough for anything the Apple might produce.

Working out your bandwidth requirements from your micro's high-resolution graphics mode can give some useful insights into what you really need from your screen. You can then go on to look at the advertised bandwidth of monitors or, in the case of a TV set, tune in to a test card to see the likely bandwidth it will accept.

The next group of things which can go wrong hinge on the nebulous concept of quality. A TV picture is a very boring thing technically. Broadcasters know that most people have mediocre TV sets and they transmit easily displayed signals to match them. Computers are different;

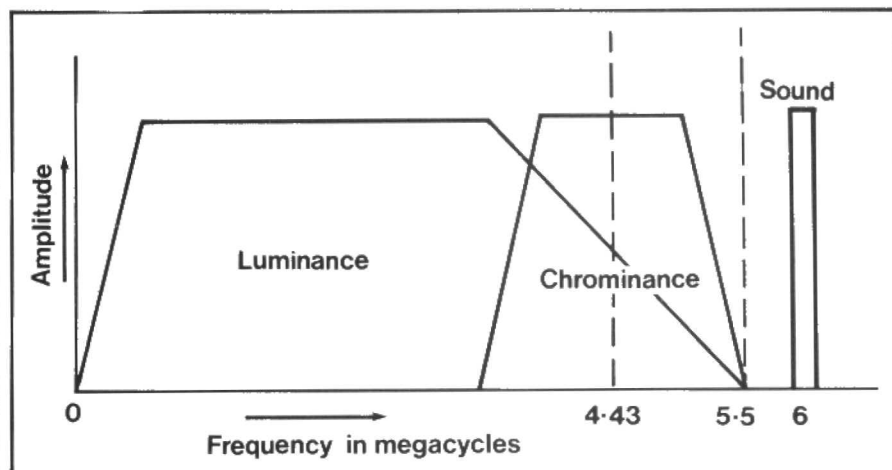


Figure 7. Allocation of bandwidth in broadcast TV.

In a colour broadcast the luminance or brightness information is transmitted in the lower frequencies. Above it is a band centred on 4.43MHz which contains the chrominance or colour information, followed by a very narrow band centred on 6MHz which contains the sound information. Because of the way a TV picture is transmitted the overlap of luminance and chrominance is not usually serious. In the overlap area the luminance signal tends to occur in clumps, and so does the chrominance signal, and these clumps tend to be in different places within the overlap area. But your computer may not be quite so well behaved as this. If it generates luminance signals over about 4MHz they are likely to be interpreted as colour information, and at 6MHz they may be interpreted as sound. A screen width of over 500 pixels or thereabouts may give problems with some colour TV sets if the input is to the aerial socket. Using an RGB input bypasses the problem because the signal does not then have to be decoded by the set.

they can send anything to the screen, and the demands they make on it are that much higher.

Switch on your computer and display some text, tuning the TV monitor to give the best possible display. Now display the same data using inverse characters and see what happens. Ideally it should be as clear and neat as before, but it may not be. The shape of the text may change, smearing slightly to the right, and if you display half a line of inverse characters the second half of the line may appear blacker than the rest of the screen, as if the bright inverse patch had cast a shadow.

TV sets are not usually called upon to display solid patches of maximum brightness, and your set's power supply may not be quite up to the job. The result is that after showing a block of maximum brightness the voltages in the set start to sag producing a reduced level of brightness immediately afterwards. Inverse text written all over the screen is most taxing of all. If anything will make the screen sag that will.

You may also notice little black dots all over the place. The modern system of having a high voltage on the video signal for black and a low voltage for white is fine for suppressing the effect of noisy spikes as long as the basic background is dark. If the screen has a white background the noise really shows badly because now you can see the black noise dots. If you want to use inverse a lot you should try to find a monitor that works the opposite way round.

Most people use a colour TV set for viewing nowadays, and they often use the same set with their computer. A colour TV

tube is exactly like a black-and-white one, except that it has three of everything. In a way, it even has three screens to give red green and blue phosphorescence.

Any problems with colour sets almost always arise from the methods used to code the colour signals. Figure 7 shows how the British PAL system of colour transmission works. Within the 8MHz bandwidth allocated to a single broadcast channel, the bottom half of the bandwidth is taken up by the luminance or brightness information, with around 4MHz devoted to the chrominance or colour information and a very narrow band to carry the sound signal. Black-and-white sets only pick up the luminance information, so colour signals can be used by everybody, whatever type of set that they have.

The luminance information must not exceed about 4MHz — if it does, it may be misinterpreted by a colour set as chrominance information. Next time you watch TV look out for high-frequency luminance information which is misinterpreted in this way. The fine detail of tweed clothing often causes problems, so people rarely wear tweeds on TV. Bright objects are also problematical. The reflection from a gold candlestick, for instance, can produce a very high-frequency edge in the picture which is mistaken for colour information to produce a sudden, vivid splodge of colour. If the object is moving to the left the problem may be even worse as it causes a Doppler effect which pushes the frequency even higher.

What bandwidth do you really need? Push 8MHz into a colour TV and the

(continued on next page)

(continued from previous page)

picture will really break up. It may even be so bad that it invades the bandwidth normally reserved for sound and your Hires graphics finish up coming out of the loudspeaker. If your computer is causing this cross-colour effect then there is not much you can do about it. All that can be said is that the computer manufacturer ought to have designed the TV output better.

If you would be happy with a monochrome picture try switching the set to monochrome and the disturbance may go away. All sets have colour-killer circuits so that if a black-and-white picture is being received it is not spoiled by colour fringing at the sharp edges of objects, where a high-frequency luminance signal exists. The colour-killer circuits are normally switched in when the set detects an absence of the chrominance information which normally occupies the higher frequencies. But any high frequencies generated by your computer in the luminance band may fool the colour-killer circuits into thinking there is some colour there. In that case your picture bursts into a glorious display of unwanted colour as everything is turned on for you.

The other problem which can arise with colour comes from the different broadcast standards around the world. All British TV sets use the Pal standard. In France they use Secam, and in the U.S. they use NTSC. So make sure you buy a computer with a Pal output otherwise it will not work with your British TV.

If you wanted to set up a small business you could go out and buy in a lot of black-and-white portable TV sets that nobody else wanted, remove the loudspeaker and the aerial socket, add a video input socket and sell the end-product as a perfectly adequate computer monitor. I am not suggesting that this is what monitor manufacturers actually do, but a monitor is really just like a monochrome TV. How good a picture it gives depends on how well it has been designed and built. When buying a monitor you must, if possible, see it in action before signing the cheque.

In theory a monitor will be better than a TV set for displaying computer output. The whole thing should be better engineered to start with and can make use of the fact that it never has to handle a TV signal. The bandwidth can be very high — up to 24MHz in some cases — and because the screen does not have the dot-matrix pattern of the colour tube it will be capable of revealing this greater detail.

Do your bandwidth sums again: 24MHz gives a frequency of 1,536 per line, so the set could resolve a basic 3,072 pixels. But remember that it will still be working at 625 lines vertically, so you will have nearly four times the resolution in the horizontal direction as in the vertical direction.

At 14MHz you can resolve about 1,500 pixels, around 20 pixels for each character on an 80-column output. Divide by two to fill in the corners on square waveforms

Guide to buying a TV or monitor

1. Work out your bandwidth requirement. If it exceeds 4MHz to draw a row of 500 pixels in high-resolution graphics then you cannot safely use a normal colour TV unless it has an RGB input. You can use a black-and-white set, or you can buy a high-resolution colour monitor. An 80-column output will normally require too high a bandwidth for a colour TV.
2. If you decide to use a normal TV try to get one with an RGB input rather than using a UHF modulator to feed into the aerial socket.
3. Always ask to see a test card displayed on a set — you should be able to resolve the finest of the vertical bars on the card, corresponding to 5.25MHz bandwidth.
4. If possible, try out the set you are thinking of with the computer you intend using. Test it with high-resolution graphics and 80-column output if you have it. Also try it with inverse text and draw blocks of maximum brightness. Look for distortion in the shape of the picture.
5. If you can, buy a set with Prestel or Ceefax on it. Displays on these channels are very similar to what your

computer may send to the set; if it cannot cope with teletext it will probably not do very well with computer output.

6. If you feel you can live without colour, then a black-and-white portable makes a very good buy and you will not be plagued by cross-colour effects. However, the power supply may be rather light, causing distortion on inverse text.

7. The safest and most expensive choice is a purpose-built monitor with 14MHz bandwidth or more. The colour of the screen phosphor for a mono monitor is a matter for personal taste.

8. The next safest choice is a black-and-white portable. Look at a test card on it and check its behaviour with inverse display. Try to get one with a video input on it.

9. A colour set with RGB input on it and one of the information channels is ideal. It should not give any problems unless you want very high resolution, but try it with your computer anyway.

10. The most difficult item is a cheap colour TV with only the aerial socket as input. A high-resolution display or an 80-column card is very likely to send it haywire. If you are thinking of buying one check it out very, very carefully first.

and you have 10 pixels per character. Because the monitor uses a video input, rather than relying on an add-on UHF modulator, the picture quality will be better than a TV anyway so it only remains to test the monitor to make sure that it is not doing anything horrible.

To test a screen connect your computer to it, using video/RGB input if possible. If you have to use an aerial socket use shielded coaxial cable because the UHF modulated signal is very susceptible to losses. Keep the cable as short as possible.

Display a screen of solid white. For instance on the Apple's 40-by-20 text screen enter:

```
FOR I = 1 TO 40*20: PRINT SPC(1);NEXT I
```

to check that the picture is rectangular at full output. Then switching to high-resolution graphics enter:

```
V = 159: H = 279: REM: V IS VERTICAL  
PIXELS, H IS HORIZONTAL
```

```
HGR  
HPLLOT 0,0 TO H,0 TO H,V TO 0,V TO 0,0
```

This is to draw a thin, white border on an overall black background and allows you to check that the screen shape remains good at low output. Go back to text again and enter:

```
NORMAL: PRINT SPC(20);  
FOR I = 1 TO 10  
INVERSE  
PRINT SPC(40);  
NORMAL  
PRINT SPC(40);  
NEXT
```

This draws a series of solid bars of white alternating with black, each extending halfway across the screen. At the middle of the screen, where a white block ends, look for a darker than usual black band following it. Is it darker than the black

band preceding white on the next line down? If it is, it shows that the low frequency response of the set — the extent to which the power sags after displaying solid patch of white — is not as good as it should be. On a normal TV the same effect is sometimes introduced deliberately because it can appear to sharpen the image up a little.

Now go back to high resolution again. Enter:

```
FOR I = 1 TO H - 1 STEP 2  
HCOLOUR = white  
HPLLOT I,0 TO I,V  
HCOLOUR = black  
HPLLOT I + 1,0 TO I + 1,V  
NEXT
```

to draw a series of vertical black-and-white bars over the surface of the screen. If there are H pixels per line, the high frequency response is H/2 cycles per line or H*7,812Hz.

If the set performs these tests without trouble then it is a fairly safe buy. If you want to examine it further using broadcast test cards you will find one transmitted on Channel 4.

Health warning

Having read this article, you may feel tempted to try messing around inside your TV set. DON'T DO IT! If you were not sure how to do it before then you do not have enough knowledge to poke around inside a TV set. A TV contains a live chassis, and the tube usually works at tens of thousands of volts. It is easily the most potentially dangerous device in the home.

Multi-million dollar industry

John Lewell assesses the computer graphics business.

IT STARTS with a distant view of the galaxy, then enters the solar system, approaches Earth, descends through the atmosphere and ends up exploring the retina of an insect's eye. It is the ultimate zoom, a computer-graphics device beloved of a number of film makers.

The computer-graphics industry itself is becoming so huge that you have to step back a long way before you can fit it all in the picture. It is estimated by Frost and Sullivan that the industry will be worth \$14.5 billion a year by 1990.

The scientific uses of computer graphics are many and wondrous, and without the new imaging techniques much scientific research would grind to a standstill. Computer graphics are used for plotting the paths of particles in high energy physics; for designing new drugs; in genetic engineering; for cartography; for enhancing the images sent back by space probes; for representing bone structures prior to surgery; for designing semiconductor chips; for modeling abstract theories; and for showing the effects of stresses on man-made objects.

Computer-aided design comes into a separate category, as it is specifically concerned with product design. CAD systems are used by engineers, industrial designers, architects, aerospace and car manufacturers, and by printed-circuit board designers.

CAD/CAM, the extension of computer-aided design, carries the process through to computer-aided manufacturing. With it a product may be manufactured automatically as soon as the design has been completed by linking the CAD system and the numerical-control machinery that is used in milling and molding processes.

All the specifications for the new product are held in a database while a three-dimensional model of the product is constructed by the designer on an electronic graphics display. When everything is ready and the computer has made all the analytical tests, modified information is passed to numerical control where lathes, mills, drills and molds are automatically set to the new specification. Presto! Your industrial robots have made a new ashtray, or space shuttle or a better mousetrap.

The category loosely labelled "business and presentation graphics" is the area of

most interest to personal-computer users, and it is the one on which we shall be concentrating. Few companies in the business-graphics area are more than 10 years old, most of them are more like 10 months old. Office graphics is very much the younger sister to scientific and CAD applications.

Within this enormous category you can find some very strange bedfellows. An animated ABC Television logo and an analytical paper graph showing the sales performance of General Motors have little in common other than the fact that both images were generated on a computer. Business and presentation graphics may exist in electronic form, on video tapes and discs, and in computer memories. Or they may be placed onto hard copies — photographic film, paper, or plastic.

Each of the three main computer-graphics categories contains a multitude of different stories but the third one, business and presentation graphics, is the most varied of all. The entire range of computing power from the massive Cray I down to the Sinclair ZX-81 can be applied in this category.

As microcomputers become more powerful they begin to play a larger role in all aspects of business and presentation graphics. Whereas research scientists and motor-car designers continue to rely on their mainframes and super-minis, the business user is finding that his other graphics needs can sometimes be met by less expensive systems. The most effective graphics hardware is currently to be found in the mid-range of computers, though you will be wise to keep one eye on those micros. The future may well belong to pint-sized processors.

Moving closer to the subject, you can see the industry in greater detail. Looked at in terms of products rather than activities, there are three main types of graphics display: the direct-view storage tube or DVST, the stroke-refresh display, and the raster-refresh display. You are more likely to encounter the stroke-refresh type in engineering and architectural design applications, and the raster-refresh type in business and presentation graphics.

The DVST was developed in the late 1960s mainly in an effort to bring down the cost of graphics displays. It forms an image

using an electron beam that moves more slowly than in other devices. The image is stored temporarily on a mesh in which phosphor is embedded. For simple applications, the DVST is still very popular since it does not require the expensive additional circuitry known as a refresh buffer.

Line-drawing displays date from the mid-1960s. They show line drawings by instructing the electron beam to connect end-points which have been digitised in the computer memory. The display processor and display-buffer memory then assemble the picture by repeatedly passing instructions to the electron gun which writes the picture on the screen.

Raster-graphics technology is closer to the operation of a normal TV set. Picture information is stored in the buffer in terms of picture elements or pixels. The horizontal scan lines of the display are composed of these individual picture elements, the whole raster being a matrix of pixels. This technology burst into prominence in the mid-1970s and is destined to become the major type of display. It appeared late on the scene because far more computing power was required to calculate the intensity and colour values of millions of pixels than to calculate just the end-point positions of vectors in a line-drawing display.

To judge the resolution of a raster display you can simply count the number of pixels; a matrix of 512 by 512 is considered average. However, when a picture is split up into discrete square elements there is a loss of resolution especially in representing diagonal lines, which tend to look like



Polaroid's VideoPrinter Model 8 colour film recorder.

staircase
anti-alias
out the
values of
solid co
illusion

The n
raster d
display
somethi
graphics
becomin
systems
actual a
modelli
approac
howeve
cramme
from th
time ca
taken t
other ha
resoluti
"Inte

finds a
The ide
a comp
the ve
Sutherl
Since t
to a po
a role
still th
both fo
dots ar

Inter
possibl
machi
keyboa
that a
porten
interfa
numbe
artists
mortal
comput
input
data
screen
tracke
switch

A r
device
while
Two s
the ba
move
instru
points
comp
as dig
Oth
pictur
stylus
stylus
sensit
indica
switch
curso
Pick
Ma
with
instar

staircases. A software technique called anti-aliasing has been developed to smooth out the jagged lines. In it the intensity values of pixels that are adjacent to areas of solid colour are adjusted to create the illusion of smooth edges.

The main difference between vector and raster displays is the ability of the raster display to show solid areas of colour, something which is essential in business graphics. Industrial designers, too, are becoming increasingly interested in raster systems because they can simulate the actual appearance of a product by solids modelling. Another plus for the raster approach is that the screen never flickers, however much visual information is crammed on to it. Vector displays suffer from this complaint because the refresh time can easily be exceeded by the time taken to write the whole image. On the other hand, raster systems rarely match the resolution of good vector systems.

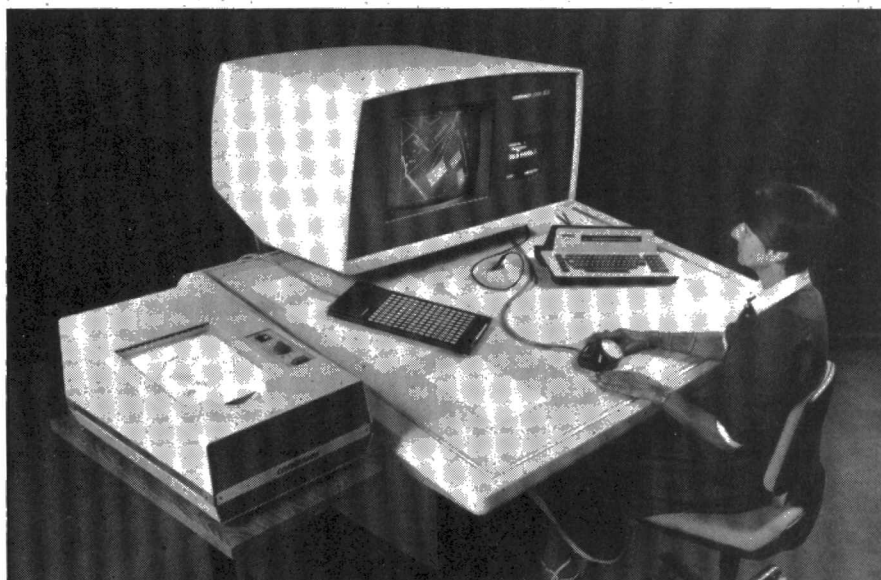
"Interactive" is a word which one often finds associated with computer graphics. The idea of a human being interacting with a computer-driven display was inherent to the very first step taken by Ivan Sutherland's Sketchpad program in 1963. Since then, the technology has been refined to a point where even voice recognition has a role to play, though most interaction is still through touch devices. They are used both for inputting and controlling the lines, dots and shapes that comprise the picture.

Interactiveness comes at many levels. It is possible to address an electronic image in machine code via an alphanumeric keyboard, but no one should really describe that as being truly interactive. What is portentously called the "human/machine interface" is dependent on having a number of convenient input devices so that artists, designers and other ordinary mortals can make pictures with the computer. Among the devices available to input or control picture information are: data tablets and stylus, touch-sensitive screens, light-pens, keyboards, joysticks, tracker balls, control dials, function switches, hand cursors and mice.

A mouse is a small hand-held locator device that can roll across a flat surface while keeping track of its own position. Two sets of tiny wheels set at right-angles in the base of the mouse register changes in movement in two dimensions. The instrument is used for locating positions of points that are to be entered into the computer memory. This process is known as digitising.

Other digitising devices for inputting pictures include the hand cursor and the stylus and tablet. The hand cursor, like the stylus, works in conjunction with a sensitive tablet. It has "cross-hairs" to indicate the point-positions. Function switches are often included on a hand-cursor to give various instructions, such as Pick or Select.

Manufacturers have really gone to town with data tablets. Sonic tablets, for instance, measure the stylus position using



The Gradis 2000, with the operator holding the cursor which is used for digitising.

strip microphones along two adjacent edges. The microphones pick up sound waves from the tip of the stylus, which emits a small electrical spark. Far more accurate — and quieter — is the tablet that has a grid of wires embedded in its surface. The co-ordinates of a point are then picked up by the stylus as variations in voltage. Yet another method is to use special material for the surface of the tablet through which electrical pulses can travel at right angles to each other. The pulses are emitted at regular intervals and so the stylus position can always be calculated.

A graphics tablet is a simulation of an artist's drawing board. Instead of seeing the image on the board you see it on a screen. Combine the tablet with a stylus which simulates a brush or a pen, and you really begin to feel like an artist. Whenever the stylus touches the tablet, a pressure-sensitive switch in the tip signals the location. A cursor may appear on the screen or, depending on the program, a pixel may be illuminated. Move the stylus and you get a row of pixels where before you had a blank screen.

Touch sensitive screens are used in some systems to give the operator a more direct contact with the electronic image. No screen cursor is then needed. There are both low-resolution and high-resolution touch screens, having 10, 50 or even 500 resolvable positions vertically and horizontally. Like the tablet, they work on several principles, including both light-wave and sound detection.

Light-pens detect light on the screen by means of a photocell located either in the pen itself or at the end of a fibre-optic pipe. They are more useful as positioning devices in line-drawing systems than as mere pointing devices in raster displays. To position images they require a tracking program in the computer software.

Keyboards are familiar to all typists and programmers. The chord keyboard, however, is a five-button device that is played like a midget's piano. You can generate 31 different instructions on it if

you are clever — but beginners should stick to the normal alphanumeric keyboard.

Joysticks and tracker balls are used for scrolling or panning the screen image. A joystick is also convenient for tumbling a wire-frame or a three-dimensional model in space. Control dials and function switches help you give instructions quickly and efficiently without having to type in more precise details on the keyboard.

The trouble with computer graphics is that people still want to carry images around with them. They like to look at them on the bus or scrutinise them in the boardroom, or project them on to a screen.

Far from decreasing the amount of printed material computers are actually increasing the quantities of paper and photographic products we consume.

Electromechanical output devices convert electronic images into hard copy. Both printers and plotters come into this category. Plotters draw while printers imprint — yet ink-jet printers imprint without even touching the paper.

The cost of electromechanical output devices range from a couple of hundred pounds for a single-pen plotter up to several hundred thousand for a sophisticated film recorder. Most of the devices have potential applications in making presentation and audio-visual graphics.

In film animation you do not have to use a film recorder: you can use a pen plotter instead. The computer-generated sequences for the American TV series "Music in Time" were created by taping an animation peg-bar on to the paper in a mechanical-arm plotter. The computer drawings, all in exact register, were later photographed and coloured by an optical camera.

Electrostatic printers are sophisticated photocopying machines, and Xerox is the leader of the pack in this field. For computer graphics they are both fast and economical, using either a matrix-writing technique or a photoconductive plate at the heart of the system. In matrix writing an

(continued on next page)

Multi-million dollar industry

(continued from previous page)

invisible image is placed on to the paper by a electrostatic charge applied by a matrix of needles. Liquid toner is then wiped on to the paper, and clings to the charged areas. In an alternative method, the image from an internal cathode-ray tube is transformed by a photoconductive plate into an electric charge on the paper.

Laser printers work in a similar way to the second electrostatic method, except that a laser beam is used instead of a CRT electron beam. These machines are very fast: the Xerox 6500 CGP prints 180 colour copies per hour, and Xerox has a machine that can produce 7,000 black-and-white charts per hour. Both electrostatic and laser can print in full colour on to a variety of media by applying carbon particles in seven colours: cyan, magenta, yellow, red, green, blue and black.

Impact and non-impact plotters tend to be relatively low-resolution devices which are well suited to providing hard-copies of raster images. Impact plotters work like sophisticated typewriters in that they use ribbons and hammers for printing. Non-impact printers work on a variety of ink-jet principles, spraying coloured inks on to paper without any physical contact between the surface being printed on and the jets. Non-impact printers are also ideal for putting images on to fragile materials.

Photographic recorders produce slides or prints from individual frames, either directly from a CRT or by collecting and reassembling the picture information. When a CRT is photographed directly the scan lines will appear in the photograph. A photographic recorder, introduced a controlled amount of blur into the picture so that the lines disappear. Sophisticated electronics are needed to match the red, green and blue CRT output with the response of photographic films.

Film recorders are similar to photographic recorders — but are around 100 times more expensive. Film recorders of this type use a special internal black-and-white single-line raster-scan CRT to display the image. Its scan is closely co-ordinated with the film transport so that the film moves a fraction of an inch after each scan line to build up an image. Colour is introduced by using a filter assembly. A powerful minicomputer analyses the image into primary colours and intensity values.

Once you have established what kind of hard copy you require, the next step is to specify the operating mode. In on-line plotting, the machine is connected to the host computer by cable. The method is fast and convenient — and expensive too because of the computer time involved. Off-line plotting makes copies from data supplied from storage media such as tapes

or discs. It can be used with a central hard-copy facility, when operators can be specially trained in this aspect of graphics.

System companies take equipment from original equipment manufacturers, design and build additional hardware, and put the whole system together as a package for specific applications and markets. They may also write or commission special software to run on their systems. Unfortunately many end-users simply do not take the trouble to shop around. If they deal with IBM for, say, data processing systems, they wait until an IBM salesman brings around a new graphics product. The alternatives are to spend weeks sifting through mountains of information, attending lengthy demonstrations, or employing a consultant.

The best approach is to hire a generalist consultant. He can suggest specialist consultants who will work with you to purchase and commission a system. Remember that computer graphics is more than just a slide-making system.

The systems of scientific and technical graphics, and for CAD/CAM need not concern us. The business and presentation graphics category contains a myriad of systems companies.

Analytical systems help us to understand the output of the computer in graphic form; presentation systems produce custom-made graphics for business presentations. Plenty of overlap exists between the two types, but the distinction is important. There is a big difference in manipulating a billion pieces of data when you make a business graph, from merely generating a computer image because you like the look of it.

Business graphics are divided into analytical and presentation systems. The products themselves can be further divided into hardware and software solutions. Hardware solutions involve purpose-built display hardware. Stand-alone systems which are usually minicomputer based are available in this category and can often be hooked into a larger data base.

The software solution is becoming increasingly popular as CPUs become more powerful. It involves the generation of graphics displays by running sophisticated software through a general-purpose computer.

Business users are demanding high resolution, ease of operation, speed, and an ability to hook into a corporate data base. A state-of-the-art device with these features will also give good performance per dollar.

When you increase the resolution of a display, for example, you immediately meet with the law of diminishing returns. As you double the number of pixels on each axis you are quadrupling their total number. Displays also feature a number of pixel layers — or planes — in the z-axis, the depth of the picture, and while this is not directly related to resolution it effects the number of colours that the display can handle.

A new feature that has been introduced

Business graphics systems

Apple Business Graphics

Turns data into graphs. Telephone Apple, Hemel Hempstead (0442) 60244.

Apple Lisa

Hard disc-based personal micro with powerful integrated graphing capabilities provided by LisaGraph. Telephone Apple Hemel Hempstead (0442) 60244.

Bit-Stik

Built around a graphics joystick with X, Y and Z directions, links to an Apple II. Telephone Robocom, 01-263 3388.

Boxer

A three-dimensional solid-modelling facility which works in conjunction with DOGS, a geometric-modelling Drawing Office Graphics System, originally developed at the University of Leeds. Telephone Pafec, Nottingham (0602) 292291.

BFS Business Graphics

In effect a word processor for business graphics, it interfaces to most popular printers and plotters and runs on the IBM PC. Imported from Cambridge, Massachusetts by Pete & Pam Computers. Telephone: Rossendale (0706) 227011.

Context MBA

Powerful spreadsheet with limited text entry, database and graphics facilities integrated together. Available for the IBM PC and Hewlett Packard Series 200

by Raster Technologies, a Massachusetts company, is a graphics display system that allows you to put the image on to either a 512-line monitor or a 1024-line monitor. With its Model One/40 you can also select the full display on the higher-resolution screen or window into the image memory in the 512 mode. The product can be described as state-of-the-art because it meets the requirements of high performance, versatility and relatively low cost.

Good software is the key to good computer graphics. The quality of the end-product — the picture or graph — is entirely shaped by the ability of the software to help you deliver an appropriate image.

Graphics software packages are a major growth area of the industry. They are available for mainframes, minis and micros. But people in the mid-range of quite expensive stand-alone mini-based systems may well find themselves in a sort of no-man's land within a few years.

Mainframe software is so powerful that once you have seen the results you will find it hard to settle for anything else. Neither is it strictly true, as some stand-alone suppliers claim, that graphics packages tie up mainframe resources at the expense of other applications. At the lower end of the market, as micros become more powerful very sophisticated software is being written for them.

The big breakthrough in software packages has been the improvement in

Model 16. Telephone The Software Rental Bank, Leighton Buzzard (0525) 373440, or Hewlett Packard (0344) 773100.

Dataplot

Business graphics package which can interface to the Supercalc spreadsheet and to Graphtext, a word-slide generating package. Runs under CP/M and MP/M. Contact Grafox, Oxford (0865) 242597.

dGraph

Versatile business graphics package which entails minimum programming and interfaces with Ashton Tate's dBase II. Telephone Fox & Geller, 01-580 5816.

Dicomed

Computer-based high-resolution colour-slide design system aimed at audio-visual producers. Dicomed copes with text, electronic drawing, digitising and freehand drawing. Eidographics then produces the final slides from your floppy discs. Contact Eidographics Ltd, 47 Marylebone Lane, London W1. Telephone: 01-486 9479.

Graforth

All-singing all-dancing graphics package written in Forth, includes a built-in music synthesizer. Telephone SBD Software, 01-948 0461.

Graphics Toolkit

Business graphics package for the ACT Sirius, interfaces with Supercalc. Telephone ACT, 021-501 2284.

Graph It

Simple graph-drawing package for Atari micros. Telephone Atari, Slough (0753) 33344.

Graphkit

Graph plotting, curve fitting and statistical analysis package for Commodore Pets. Telephone Commodore Information Centre, Slough (0753) 79292.

Graph'n'Calc

Graph-drawing program with its own modest spreadsheet, for the IBM PC. Imported from Santa Cruz, California by Pete & Pam. Telephone: Rossendale (0706) 227011.

Lotus 1-2-3

Spreadsheet with a powerful graphic extension built in. Telephone Planning Consultancy, 01-839 3143.

Peachtree Graphics Language

Interactive graphics programming language which runs under CP/M and MP/M, and interfaces to Peachcalc/Magicalc and Peachtext/Magic Wand. Telephone Peachtree, Maidenhead (0628) 32711.

Perq

Amazing mouse-driven graphics system for draughting and CAD/CAM applications. Telephone ICL, Infopoint, 01-788 7272.

P.L. Graphics

Digitising and drawing system based on

the BBC Model B and suitable for schools and small businesses. Telephone B S Dollamore, Burton-on-Trent (0283) 217905.

StarGraphics II

35mm. colour-slide design system based on the Apple II micro. Myriad produce the final slides from your discs. Contact Myriad, 106 Hampstead Road, London NW1. Telephone: 01-380 0191.

Sub-Logic Graphics Package

For displaying three dimensional scenes on a two dimensional display. Telephone Pete & Pam, Rossendale (0706) 227011.

Utopia Graphics Table System

Provides 64 colours, 40 brush shades and pen-controlled editing.

Vectrix VX Series

CAD/CAM and business graphics system which includes NEC chips and an Intel 8088, and links to various personal computers including the IBM PC, Sirius, Osborne and Hewlett Packard and Apple. The VX-384 can display up to 512 colours from a palette of 16.8 million. Imported from Greensboro, North Carolina by Sintrom Electronics. Telephone: Reading (0734) 875464.

VisiTrend/Plot

Converts data from VisiCalc and other Visi products into business graphics. Telephone Rapid Recall, (0494) 38525.

user-friendliness. This is a marketing breakthrough because the people who really need graphics are rarely skilled in computing. Yet the most successful cases of business graphics implementation have been where hundreds and even thousands of company employees have been trained to operate a system. No longer is computer graphics an arcane and mysterious art. The new techniques are available to everyone.

First-rate graphics software is now almost affordable by everyone. VisiPlot, to run on an Apple computer, is priced at around £177. At these rates, the use of computer graphics will become almost universal in small businesses and perhaps

even in the home.

Word processors can be converted into graphics workstations with the addition of appropriate software. Writing a software package, however, requires highly-skilled programmers. The business graphics packages offered by Apple Computers took 200 man-years to develop — and woman-years no doubt.

The academic interest in artificial intelligence is influencing new approaches to computer graphics. Eventually even the most advanced scientific thought reaches the businessman in one form or another. One phenomenon of particular interest to students of artificial intelligence is pattern

recognition. We do not fully understand why or how we instantly recognise, say, the face of a friend in a crowd of people, when everyone in the crowd has two eyes, two ears and a nose in approximately the same places. It prompts the question, can patterns be generated and developed to help the businessman recognise the friendly face of his sales statistics?

Over the past decade we have survived a blitz of multi-image shows which have helped pave the way for an appreciation of pattern recognition as a business tool. Today's incredible growth of computer graphics is both a symptom and a cause of this new development of human skills.

Business graphics

Our world's economy, depicted in colourful business graphics, looks pretty unhealthy. Today you can see all those complex statistics at a glance, arranged for you by the computer in graphic format. Gone is the time when economists could pretend to be deaf when asked which way the wind was blowing. But if charts and graphs show us a sorry picture of the world recession at least the world of business graphics is booming, at a rate of around 70 percent a year.

Graphics Software, Inc., an Oregon-based business graphics company, makes a startling claim, "Mainframe quality graphics can now be replicated on your mini/microcomputer." The claim will bear scrutiny providing you read it carefully. The company is not suggesting that an Apple II can perform the same service as a big mainframe complete with Tell-a-Graf software. Rather, it is saying that its product, GSS Plot, will lift micro graphics to a new level of flexibility. The software

will run on most mini and microcomputers, and it is fully device independent in regard to CRTs and plotters.

This particular software package has been designed specifically for applications software developers. GSS Plot contains all the necessary computer instructions to prepare presentation quality line graphs, bar charts, scatter charts and other types of display. For example, a user need call only four related sub-routines and supply 13 parameters in order to generate a complex pie chart. Without the GSS Plot subroutines you would have to write a 100 lines of code and make all the necessary tests before being able to generate similar graphics.

Since CP/M has become the favoured operating system of most personal-computer manufacturers, GSS has signed a deal with the CP/M originators, Digital Research Inc., and now offers GSS Plot in CP/M compatible form. This co-operation could have a significant impact on the future of business graphics,

(continued on next page)

Multi-million dollar industry

(continued from previous page)

enabling micro users to add a true graphics capability to their machines. Tom Clarkson, president of GSS, says that the agreement, "will significantly assist our ability to make standardised graphics software available to micro and minicomputer users."

Another company to watch in this field is Graphic Communications Inc., of Waltham, Massachusetts. Its President, Randall E Wise, is a strong believer in what he calls the software solution. He says, "There are hardware solutions to stand-alone graphics and there are software solutions to stand-alone graphics. We have chosen the software solution, currently operating on the HP-8587. That concept works on an IBM Personal Computer as an application. And powerful 16-bit computers are coming along with very good software, which can very nearly duplicate the capabilities that other stand-alone systems are offering with their hardware solutions".

Special hardware is always more expensive than general-purpose software. Randall Wise suggests that the companies who are offering expensive hardware for business graphics will find themselves fighting a losing battle on prices. "I don't know how they are going to react to the new software that will duplicate their capability for a few thousand dollars."

Business graphics are often required in slide or overhead transparency format. Polaroid instant film technology has made a big impact on this market with several manufacturers incorporating instant film cameras in their systems. Polaroid has several products, in particular, the Videoprinters Models 4 and 8. "We can now bridge the gap between electronics and film," says James Hartnett, Marketing Manager of Polaroid's Professional Film Products Division. "Previously, film had not been appropriately matched in phosphor response to proper exposures of red, green, or blue levels. In the Model 8 it is possible to optimise the two — and get the best result."

The Model 8 Videoprinter is a microprocessor-controlled display-driven device, producing 8in. by 10in. instant colour overhead transparencies. "It is very state-of-the-art in digital input and information gathering," says Hartnett. The Model 4 is intended for 4 by 5 formats and smaller, including the new instant-process 35mm. film that Polaroid is launching later in 1982.

One of the first micro-based graphics systems came from Cromemco, a company that is better known for its CAD/CAM hardware. The Z-2H graphics system is based on the Z-80 chip, and has recently been upgraded to include the Motorola 68000. This allows the display of a 1,000 by 1,000 matrix, putting it just into the high-resolution bracket. Cromemco has supported its system with two software packages, Slidemaster and Fontmaster. With Slidemaster and a graphics tablet a user can choose from 75 design functions that are displayed on both screen and tablet menus. The package is intended for presentation graphics, and includes a carousel mode that allows you to call up an image sequence as though you were controlling a slide projector. Fontmaster lets you design your own lettering or special characters such as scientific notation.

Excellent software for both the Apple II and III has been developed by Business and Professional Software of Cambridge, Massachusetts. Recent packages from this company, marketed by Apple themselves, go beyond the presentation graphics of its earlier software. The two-diskette package, now called Apple Business Graphics, allows you to create colour graphic

representations of data using English language commands. For instance, sales projection data can be retrieved from a VisiCalc program and automatically displayed as bar, line or pie charts.

Like the Cromemco system, the Apple Business Graphics package can make the computer function like a slide projector. It requires the addition of a new product called Screen Director, two diskettes that come with a Kodak hand-held projector controller, which plugs into the game-slot on the Apple. David Solomont, President of BPS, says, "Apple Business Graphics allows the user to create and store graphic images. Screen Director retrieves and displays them on video monitors for presentation." It also lets you create hard copies of a whole tray of images on many brands of dot matrix printer, including the IDS Paper Tigers, Anadex 9000s, Epson MX line and Apple Silentype. The IDS Prism will produce hard copy in colour.

With the low cost of Apple software and the relatively high cost of colour slide-making systems an attractive solution is to make use of a hard-copy bureau service. Comshare has tried out its Target Image Maker on the Apple. Users of the system would be able to create charts on their in-house computer and then download them to a Comshare facility for production of the slides. At this point the Post Office takes over, the postman brings you the slides in the mail. Perhaps this deflates



The Apple Business Graphics package makes the computer function like a slide projector.

some of the high-tech magic that surrounds the creation of electronic images, but it is also the thinking behind the Cornerstone/Clear Light Stargraphics operation.

If business graphics becomes as simple and as inexpensive as many experts predict, we shall be knee-deep in graphs and charts before the decade is over. Manufacturers are already treating the subject as though images will be manipulated with the ease of words in word-processing. Another name will have to be found for this technique, since image processing means something quite different — image enhancement.

Hinting at the shape of the office of the future, Hewlett-Packard can now proudly show off its Merged Text and Graphics system. This will actually produce illustrated business correspondence, among other applications. Perhaps we shall eventually be able to dispense with words altogether and communicate with each other entirely in pictures. This, of course, will only deepen the world recession. The left side of our brains will be redundant, while the right side will be on strike — demanding extra pay for extra work.

Geometrical

THE STANDARD METHOD of plotting lines on computer-graphics systems is very similar to the way a pen and paper is used.

The main difference is that the computer is a comparatively simple machine and can normally only draw straight lines. To draw a circle, for example, the computer has to divide the curve up into short straight-line segments which it then plots individually.

Even drawing lines requires a lot of software. Anything less straightforward calls for a library of routines. Often the programmer must start from scratch using only a basic line-plotting routine supplied with the system.

The listing provides a library of useful plotting routines written in Basic which should run on any machine. Only one routine needs to be written to interface the library with most line-plotting systems, and usually it only needs to be a line or two in length.

The interface routine is at line 1000 and is used by all the other plotting routines in the library. It moves the cursor from the current position to a new position specified with a pair of X,Y co-ordinates in the parameters. A third parameter is used to specify the mode in which the cursor is to be moved. If it is zero the cursor is simply moved to the new position. If it is non-zero a line is also drawn from the old to the new position. The value of the parameter may be used to specify the colour and/or intensity of the line to be drawn.

In the library it is assumed that only one colour is available for the line drawing. You could use a global variable to specify the colour to be used for plotting if your system allows it. The routine at line 1000 must then use this variable to plot the correct colour.

All the routines, apart from the basic plotting routine and the conversion routines, need to be supplied with a starting position offset X0 and Y0 from which plotting will commence. The variables T0 to T9 may be corrupted by the library routines and thus should not be used to hold global values.

None of the routines in the library contain line-number references using Goto or GOSUB statements, apart from line 1000 for the basic line-plotting routine. You can easily relocate the code to different line numbers, if you wish, without changing the code itself, provided the Plot routine is on line 1000.

The library should be stored as a single file. When a new program which uses some of the library routines is to be written, the library itself should first be loaded. Unused routines may be deleted and the program typed in on the keyboard. The program and library routines may then be renumbered as required before being saved as another file.

Interface routine

Plot, line 1000. Plots from current position to new position. The only routine in the library which needs to be

written for a specific system. Most of the other routines in the library call this routine. Input parameters are:

X — X co-ordinate of new position
Y — Y co-ordinate of new position
M — mode of plot; 0 to move without drawing line, 1 to draw line to new position

Line-plotting routines

Draw, line 1100. Draws a line between two positions. Useful when the two end co-ordinates of the line are known and the current position of the cursor is at neither of them. Input parameters are:

X0 — starting X co-ordinate
Y0 — starting Y co-ordinate
L1 — finishing X co-ordinate
L2 — finishing Y co-ordinate

Line, line 1200. Draws a line from a position. Should be used when the starting co-ordinates and the relative position of the finishing co-ordinates are known. Input parameters are:

X0 — starting X co-ordinate
Y0 — starting y co-ordinate
L1 — increment in X direction for final position
L2 — increment in Y direction for final position

Radius, line 1300. Draws radius of a circle. Useful when the angle and the length of the lines are known rather than the X,Y co-ordinates of the end of the line. Input parameters are:

X0 — starting X co-ordinate

Geometrical plotting.

```
1000 REM "PLOT",X,Y,M
1010 REM
1020 REM (C) J.P.BOWEN, OCTOBER 1982
1030 REM
1040 REM *****
1050 REM * Machine dependent code *
1060 REM *****
1070 REM
1080 REM
1090 RETURN
1100 REM "DRAW",X0,Y0,L1,L2
1110 M=0
1120 X=X0
1130 Y=Y0
1140 GOSUB 1000
1150 M=1
1160 X=L1
1170 Y=L2
1180 GOSUB 1000
1190 RETURN
1200 REM "LINE",X0,Y0,L1,L2
1210 M=0
1220 X=X0
1230 Y=Y0
1240 GOSUB 1000
1250 M=1
1260 X=X0+L1
1270 Y=Y0+L2
1280 GOSUB 1000
1290 RETURN
1300 REM "RADIUS",X0,Y0,L,A
1310 T0=A*PI/180
1320 M=0
1330 X=X0
1340 Y=Y0
```

```
1350 GOSUB 1000
1360 M=1
1370 X=X0+L*COS(T0)
1380 Y=Y0+L*SIN(T0)
1390 GOSUB 1000
1400 RETURN
1410 REM "FIGURE",X0,Y0,L1,L2,N1,N2,
      X(?),Y(?)
1420 M=0
1430 X=X0+L1*X(N1)
1440 Y=Y0+L2*Y(N1)
1450 GOSUB 1000
1460 M=1
1470 FOR T0=N1+1 TO N2
1480 X=X0+L1*X(T0)
1490 Y=Y0+L2*Y(T0)
1500 GOSUB 1000
1510 NEXT T0
1520 RETURN
1530 REM "POLYGON",X0,Y0,L1,A,N,N1
1540 T0=A*PI/180
1550 T1=2*PI/N1
1560 M=0
1570 X=X0
1580 Y=Y0
1590 GOSUB 1000
1600 M=1
1610 FOR T2=1 TO N
1620 X=X0+L*COS(T0)
1630 Y=Y0+L*SIN(T0)
1640 T0=T0+T1
1650 GOSUB 1000
1660 NEXT T2
1670 IF N(N1) THEN RETURN
1680 X=X0
1690 Y=Y0
1700 GOSUB 1000
```

```
1710 RETURN
1720 REM "SQUARE",X0,Y0,L,A
1730 L1=L
1740 L2=L
1750 N=4
1760 REM "RECTANGLE",X0,Y0,L1,L2,A,N
1770 T0=A*PI/180
1780 T1=COS(T0)
1790 T2=SIN(T0)
1800 M=0
1810 X=X0
1820 Y=Y0
1830 GOSUB 1000
1840 M=1
1850 IF N<1 THEN RETURN
1860 X=X0+L1*T1
1870 Y=Y0+L1*T2
1880 GOSUB 1000
1890 IF N<2 THEN RETURN
1900 X=X0+L2*T1
1910 Y=Y0+L2*T2
1920 GOSUB 1000
1930 IF N<3 THEN RETURN
1940 X=X0+L2*T2
1950 Y=Y0+L2*T1
1960 GOSUB 1000
1970 IF N<4 THEN RETURN
1980 X=X0
1990 Y=Y0
2000 GOSUB 1000
2010 RETURN
2020 REM "TRIANGLE",X0,Y0,L1,L2,A,N
2030 T0=A*PI/180
2040 T1=COS(T0)
2050 T2=SIN(T0)
2060 M=0
2070 X=X0
```


plotting

Y0 — starting Y co-ordinate
L — length of radius
A — angle of radius

Geometrical plotting

Figure, line 1410. Draws an irregular figure. The co-ordinates of the vertices of the figure are passed as arrays in the parameters X and Y which must be set up before the routine is called. For example:

```
100 N1 = 1
110 INPUT N2
120 DIM X(N2), Y(N2)
130 FOR I = N1 TO N2
140 INPUT X(I), Y(I)
150 NEXT I
160 X0 = 0
170 Y0 = 0
180 L1 = 1
190 L2 = 1
200 GOSUB 2100: REM "FIGURE"
```

The parameters N1 and N2 specify the range of the arrays to be used; in this example the entire array is used. The parameters X0, Y0, L1 and L2 may be used to offset and scale the figure. Input parameters are:

X0 — offset in the X direction
Y0 — offset in the Y direction
L1 — scaling factor in the X direction
L2 — scaling factor in the Y direction
N1 — first array subscript to be used
N2 — last array subscript to be used
X — array containing X co-ordinates
Y — array containing Y co-ordinates

Polygon, line 1530. Draws a regular polygon. As well as the number of sides of the polygon, the number of sides to

be drawn is also specified. The polygon may be drawn at any angle to the horizontal. Input parameters:

X0 — starting X co-ordinate
Y0 — starting Y co-ordinate
L — length of side
A — angle of first side; normally zero
N — number of sides to be drawn; normally equal to N for a complete polygon
N1 — number of sides; must be three or more

Square, line 1720. Draws a square. Drops through to the rectangle routine with the correct parameters to draw a square. Input parameters are:

X0 — starting X co-ordinate
Y0 — starting Y co-ordinate
L — length of sides of square
A — angle of first side; normally zero for horizontal square

Rectangle, line 1760. Draws a rectangle. The size of the base and height, number of sides to be drawn and angle to the horizontal must be specified. Input parameters:

X0 — starting X co-ordinate
Y0 — starting Y co-ordinate
L1 — length of base of rectangle
L2 — height of rectangle
A — angle of first side; normally zero for horizontal rectangle
N — number of sides to be drawn; normally four for complete rectangle

Triangle, line 2020. Draws an isosceles

Jonathan Bowen presents a library of versatile Basic procedures which add a graphics-handling capability to your micro.

triangle. The figure is drawn clockwise so that if only two sides are drawn then they are symmetrical. The angle from the horizontal may be varied. Input parameters are:

X0 — starting X co-ordinate
Y0 — starting Y co-ordinate
L1 — length of base of triangle
L2 — height of triangle
A — angle of base
N — number of sides to be drawn; normally three for complete triangle

Circular curve plotting

Arc, line 2240. Draws a circular arc. The centre of arc and the radius must be specified, together with the starting and finishing angles from the horizontal. The number of straight-line segments needed to make up the arc is calculated automatically and then the Segment Arc routine is used. The segment number calculation assumes a plotting area of a few hundred pixels in each direction. If this is not the case on a particular system, then the division factor — 3 in this case — may need to be altered to obtain satisfactory results. For example:

X0 — X co-ordinate of centre of arc
Y0 — Y co-ordinate of centre of arc
L — length of arc radius
A1 — starting angle
A2 — finishing angle

Segment Arc, line 2260. Draws a segmented arc of a circle. The parameters are as for the Arc subroutine

(continued on next page)

```
1460 Y=Y0
1470 GOSUB 1000
1500 M=1
1510 IF N<1 THEN RETURN
1520 X=X0+L1*T1/2-L2*T2
1530 Y=Y0+L1*T2/2+L2*T1
1540 GOSUB 1000
1550 IF N<2 THEN RETURN
1560 X=X0+L1*T1
1570 Y=Y0+L1*T2
1580 GOSUB 1000
1590 IF N<3 THEN RETURN
1600 X=X0
1610 Y=Y0
1620 GOSUB 1000
1630 RETURN
1640 REM "ARC",X0,Y0,L,A1,A2
1650 N=20+INT(L*ABS(A2-A1)/1000)
1660 REM "SEGMENT ARC",X0,Y0,L,A1,A2,N
1670 T0=A1*PI/180
1680 T1=A2*PI/180
1690 T2=(T1-T0)/N
1700 M=0
1710 X=X0+L*COS(T0)
1720 Y=Y0+L*SIN(T0)
1730 GOSUB 1000
1740 M=1
1750 FOR T3=2 TO N
1760 T0=T0+T2
1770 X=X0+L*COS(T0)
1780 Y=Y0+L*SIN(T0)
1790 GOSUB 1000
1800 NEXT T3
1810 X=X0+L*COS(T1)
1820 Y=Y0+L*SIN(T1)
1830 GOSUB 1000
1840 RETURN
```

```
2450 REM "CIRCLE",X0,Y0,L
2460 N=20+INT(L/3)
2470 REM "SEGMENT CIRCLE",X0,Y0,L,N
2480 T0=2*PI/N
2490 M=0
2500 X=X0+L
2510 Y=Y0
2520 GOSUB 1000
2530 M=1
2540 T1=0
2550 FOR T2=2 TO N
2560 T1=T1+T0
2570 X=X0+L*COS(T1)
2580 Y=Y0+L*SIN(T1)
2590 GOSUB 1000
2600 NEXT T2
2610 X=X0+L
2620 Y=Y0
2630 GOSUB 1000
2640 RETURN
2650 REM "DOT",X0,Y0
2660 X=X0
2670 Y=Y0
2680 M=0
2690 GOSUB 1000
2700 M=1
2710 GOSUB 1000
2720 RETURN
2730 REM "DOT GRID",X0,Y0,L1,L2,N1,N2
2740 T0=L1/(N1-1)
2750 T1=L2/(N2-1)
2760 Y=Y0
2770 FOR T2=1 TO N2
2780 X=X0
2790 FOR T3=1 TO N1
2800 M=0
2810 GOSUB 1000
```

```
2820 M=1
2830 GOSUB 1000
2840 X=X+T0
2850 NEXT T3
2860 Y=Y+T1
2870 NEXT T2
2880 RETURN
2890 REM "DOT LINE",X0,Y0,L,A,N
2900 T0=A*PI/180
2910 T1=L*COS(T0)/(N-1)
2920 T2=L*SIN(T0)/(N-1)
2930 X=X0
2940 Y=Y0
2950 FOR T3=1 TO N
2960 M=0
2970 GOSUB 1000
2980 M=1
2990 GOSUB 1000
3000 X=X+T1
3010 Y=Y+T2
3020 NEXT T3
3030 RETURN
3040 REM "DOTS",X0,Y0,L1,L2,N
3050 FOR T0=1 TO N
3060 X=X0+L1*RND(1)
3070 Y=Y0+L2*RND(1)
3080 M=0
3090 GOSUB 1000
3100 M=1
3110 GOSUB 1000
3120 NEXT T0
3130 RETURN
3140 REM "ARROW",X0,Y0,L,L1,A,A1,N
3150 T0=A*PI/180
3160 T1=A1*PI/180
3170 T2=X0+L*COS(T0)
3180 T3=Y0+L*SIN(T0)
```

(listing continued on next page)

(continued from previous page)

except that the number of straight-line segments in the arc must also be given:

X0 — X co-ordinate of centre of arc
Y0 — Y co-ordinate of centre of arc
L — length of arc radius
A1 — starting angle
A2 — finishing angle
N — number of segments in arc; must be 2 or more

Circle, line 2450. Draws a circle. the parameters are as for the Arc routine except that the starting and finishing angles need not be specified. The number of straight-line segments is calculated automatically before the Segment Circle routine is used. As with the Arc routine, the segment-number calculation assumes a plotting area of a few hundred pixels in each direction. If this is not the case, then the division factor — 1,080 in the library routine shown — may need to be adjusted to obtain satisfactory results. Input parameters are:

X0 — X co-ordinate of centre of circle
Y0 — Y co-ordinate of centre of circle
L — length of circle radius

Segment Circle, line 2470. Draws a segmented circle. The parameters are as for the Circle subroutine except that the number of straight-line segments to be used must also be given:

X0 — X co-ordinate of centre of circle
Y0 — Y co-ordinate of centre of circle
L — length of circle radius
N — number of straight-line segments

Dot-plotting routines

Dot, line 2650. Draws a dot. A similar calling sequence is used by the rest of the routines in this section. Input parameters:

X0 — X co-ordinate of dot
Y0 — Y co-ordinate of dot

Dot Grid, line 2730. Draws a rectangular grid of dots. The size and the number of dots in each direction must be specified. Input parameters are:

X0 — X co-ordinate of bottom left-hand corner of grid
Y0 — Y co-ordinate of bottom left-hand corner of grid
L1 — length of rectangle in X direction
L2 — length of rectangle in Y direction
N1 — number of dots in X direction; must be two or more
N2 — number of dots in Y-direction; must be two or more

Dot Line, line 2890. Draws a line of dots. The parameters are as for the Radius subroutine except that the number of dots to be plotted must also be specified:

X — X co-ordinate of start of line
Y — Y co-ordinate of start of line
L — length of line
A — angle of line
N — number of dots in the line; must be two or more

Dots, line 3040. Plots random dots in a rectangle. The number of dots must be given. The routine assumes that the function RND (1) returns a random number between 0 and 1. If not, it will need to be adjusted accordingly. Input parameters are:

X0 — X co-ordinate of bottom left-hand corner of rectangle
Y0 — Y co-ordinate of bottom right-hand corner of rectangle
L1 — length of rectangle in X direction
L2 — length of rectangle in Y direction
N — number of dots to be plotted

General-purpose plotting

Arrow, line 3140. Draws an arrow. The length and angle of the shaft and head must be specified. The head may be either open or closed. Input parameters are:

X0 — starting X co-ordinate
Y0 — starting Y co-ordinate
L — length of shaft
L1 — length of head
A — angle of shaft
A1 — angle of head from shaft
N — 0 for open head; 1 for closed head

Dashes, line 3390. Draws a dashed line. The parameters are as for the Radius subroutine except that the number of dashes and the ratio of dash to space between the dashes must also be given:

X0 — X co-ordinate of start of line
Y0 — Y co-ordinate of start of line
L — length of line
A — angle of line
N — number of dashes
N1 — ratio of dash to space between dashes

Grid, line 3660. Draws a rectangular grid. Parameters are as for the Dot Grid routine, this time the grid is drawn with solid lines.

X0 — X co-ordinate of bottom left-hand corner of grid
Y0 — Y co-ordinate of bottom left-hand corner of grid
L1 — length of grid in X direction
L2 — length of grid in Y direction
N1 — number of divisions in X direction
N2 — number of divisions in Y direction

Hatch, line 3890. Hatches in a rectangle. The number of lines used and the angle of the hatching are specified as parameters. For horizontal lines the angle is specified as zero; 90 degrees gives vertical lines.

X0 — X co-ordinate of bottom left-hand corner of rectangle
Y0 — Y co-ordinate of bottom left-hand corner of rectangle
L1 — length of rectangle in X direction
L2 — length of rectangle in Y direction
A — angle of hatching; between 0 and 180 degrees
N — number of hatching lines

(listing continued from previous page)

```
3190 M=0
3200 X=X0
3210 Y=Y0
3220 GOSUB 1000
3230 M=1
3240 X=T2
3250 Y=T3
3260 GOSUB 1000
3270 X=T2-L1*COS(T1-T0)
3280 Y=T3-L1*SIN(T1-T0)
3290 GOSUB 1000
3300 IF N=0 THEN M=0
3310 X=T2-L1*COS(T1+T0)
3320 Y=T3-L1*SIN(T1+T0)
3330 GOSUB 1000
3340 M=1
3350 X=T2
3360 Y=T3
3370 GOSUB 1000
3380 RETURN
3390 REM "DASHES",X0,Y0,L,A,N,N1
3400 T0=A*PI/180
3410 T1=COS(T0)
3420 T2=SIN(T0)
3430 T3=L/(N*(N1+1)-1)
3440 T4=T3*T1
3450 T5=T3*T2
3460 T6=N1*T4
3470 T7=N1*T5
3480 M=0
3490 X=X0
3500 Y=Y0
3510 GOSUB 1000
3520 FOR I=1 TO N
3530 M=1
3540 X=X+T6
```

```
3550 Y=Y+T7
3560 GOSUB 1000
3570 M=0
3580 X=X+T4
3590 Y=Y+T5
3600 GOSUB 1000
3610 NEXT I
3620 X=X0+L*T1
3630 Y=Y0+L*T2
3640 GOSUB 1000
3650 RETURN
3660 REM "GRID",X0,Y0,L1,L2,N1,N2
3670 T0=L1/N1
3680 X=X0
3690 FOR T1=0 TO N1
3700 M=0
3710 Y=Y0
3720 GOSUB 1000
3730 M=1
3740 Y=Y0+L2
3750 GOSUB 1000
3760 X=X+T0
3770 NEXT T1
3780 T0=L2/N2
3790 FOR T1=0 TO N2
3800 M=0
3810 X=X0+L1
3820 GOSUB 1000
3830 M=1
3840 X=X0
3850 GOSUB 1000
3860 Y=Y0+T0
3870 NEXT T1
3880 RETURN
3890 REM "HATCH",X0,Y0,L1,L2,A,N
3900 T0=N+1
3910 T1=A
3920 IF (T1<0) OR (T1)=180 THEN T1=0
```

```
3930 IF T1<90 THEN T2=ABS(TAN(T1*PI/180))
3940 T3=L1/T0
3950 IF (T1<0) AND (T1<90) THEN T3=
(L1+L2/T2)/T0
3960 IF T1<90 THEN T3=-T3
3970 IF T1<90 THEN T4=(L1*T2+L2)/T0
3980 T5=0
3990 IF T1<90 THEN T5=INT(L2/T4)
4000 T6=X0
4010 IF T1<90 THEN T6=X0+L1
4020 T7=X0
4030 IF T1=90 THEN T7=X0+L1
4040 T8=Y0+L2
4050 FOR T2=1 TO N
4060 T9=T0-T2
4070 X=T6
4080 IF T2>T5 THEN X=T7-T9*T3
4090 Y=T8
4100 IF T2<T5 THEN Y=Y0+T2*T4
4110 M=0
4120 GOSUB 1000
4130 X=T7
4140 IF T9>T5 THEN X=T6+T2*T3
4150 Y=Y0
4160 IF T9<T5 THEN Y=T8-T9*T4
4170 M=1
4180 GOSUB 1000
4190 NEXT T2
4200 RETURN
4210 REM "PI".PI
4220 PI=4*ATN(1)
4230 RETURN
4240 REM "RADIAN".A
4250 A=A*PI/180
4260 RETURN
4270 REM "DEGREE".A
4280 A=A*180/PI
4290 RETURN
```


Colouring blocks

Many shapes are difficult to fill using the BBC Micro's triangle-drawing facility.

John Dallman explores the possibilities offered by the new BBC ROM.

ONE OF THE FEW serious gaps in the graphics abilities of the BBC Micro was, until recently, the lack of a general-purpose facility for producing solid blocks of colour. While the built-in triangle-drawing facility is very useful, many shapes cannot easily be produced from triangles.

It is sometimes useful to be able to fill in a shape which has been drawn, but whose boundaries are not known in a convenient form for the triangle-filling routines. Many kinds of graphs, shapes sketched by hand on the screen using a light-pen and landscapes for games are all difficult to fill using triangles.

What was needed was an intelligent graphics routine that could be given a starting point inside any closed shape and then fill it with a specified colour. Some microcomputers already have software which will do this job.

Now Acorn Computers has announced that some extra area-filling routines have been added to the BBC operating system. At first sight, they do not seem very impressive but a closer inspection shows that they are the basic operations for a very powerful algorithm which is well adapted to small systems.

The new operating system calls fill a horizontal row of pixels with a specified colour. They are implemented as two new groups of Plot operations, and may therefore be used directly from Basic. An additional Oswald enquiry call has also been added, and can be accessed from Basic through the Call statement.

The techniques based on these new routines will only work on a BBC Micro with a Series 1 operating system ROM fitted. Owners of machines with the earlier version will find new ROMs available at BBC dealers and through the Beebug user group. The recursive techniques described by listing 3 will work on any BBC machine and, with modifications, on any system that allows recursive programming.

The new Plot routines are available from Basic with the statement

PLOT K,X,Y

where K is the Plot option — 72 to 79 in

this case — and X and Y are the co-ordinates of a point on the screen. When used, these routines start at the pixel specified by X and Y and search leftwards and rightwards along the same row for a pixel not in the current background colour.

The search stops when it reaches the left-most and right-most pixels that can be reached from the starting point without crossing any pixel not set to the background colour.

The system variables holding the last two positions of the graphics cursor are then set to the co-ordinates of these points, and a line is drawn between them. Table 1 shows the exact meanings of X and K and

the types of line drawn for different Plot options.

Plot options 88 to 95 work in a similar way but expect to be given a starting point not set to the background colour. These commands search for the last point that is not set to the background colour, moving away from the starting point. This point and the starting point are used as the new values for the last two cursor positions and a line may be drawn between them. Table 2 gives details of the individual commands.

In listing 1 a square is drawn and then filled in using Plot 77. A line is then drawn in the background colour, and an area to the left of it is filled in with a different colour. The program uses mode 5 so that individual pixels are clearly visible on the screen.

The Step 4 statements in lines 230 and 430 are important. The area-filling routines work in terms of physical pixels on the screen when searching, not 1,280-by-1,024 set of users co-ordinates. On the BBC Micro, there are only 256 pixels on the vertical axis of the screen, so the Step 4 prevents duplication of an operation after a row has been filled.

The routines work strictly in terms of logical colours and ignore any alterations of the default colours by VDU 19 statements. If you happen to have two logical colours set to the same physical colour and are using one of them as the background colour the routines will be able to tell the difference even though none is visible on the screen.

Selecting a new background colour with the GCol statement will not change the background until the screen is cleared. However, any of these new Plot statements used between selecting a new background colour and clearing the screen will treat the newly selected value as the background colour when deciding if a given pixel is set to the background colour or not.

The new operating-system call has the form

OSWORD 13 (&OD)

It is used as an enquiry call, returning the

Listing 1.

```

10 REM Program 1 - demonstration
20 REM of new BBC PLOT routines
30 REM in MOS 1.2.
40 REM By J.G.Dallman, June 1983.
50
60 MODE 5
70
80 REM Draw a shape to be filled in:
90 REM a square 400 x 400, centered
100 REM at 400,500.
110
120 MOVE 400,500
130 DRAW 800,300
140 DRAW 800,700
150 DRAW 400,700
160 DRAW 400,300
170
180 PROCpause(10)
190
200 REM Fill the square in in yellow.
210
220 GCOL0,2
230 FOR YX = 300 TO 700 STEP 4
240 PLOT 77,600,YX
250 NEXT
260
270 PROCpause(10)
280
290 REM Draw a line in the back-
300 REM ground colour (black).
310
320 GCOL0,0
330 MOVE 600,400
340 DRAW 750,500
350 DRAW 600,600
360
370 PROCpause(10)
380
390 REM Fill rightwards to that line,
400 REM in red.
410
420 GCOL0,1
430 FOR YX = 400 TO 600 STEP 4
440 PLOT 93,500,YX
450 NEXT
460 END
470 DEF PROCpause(secs)
480 REM Wait for 'secs' seconds.
490 LOCAL t
500 t = TIME
510 REPEAT
520 UNTIL TIME > t + 100*secs
530 ENDPROC

```

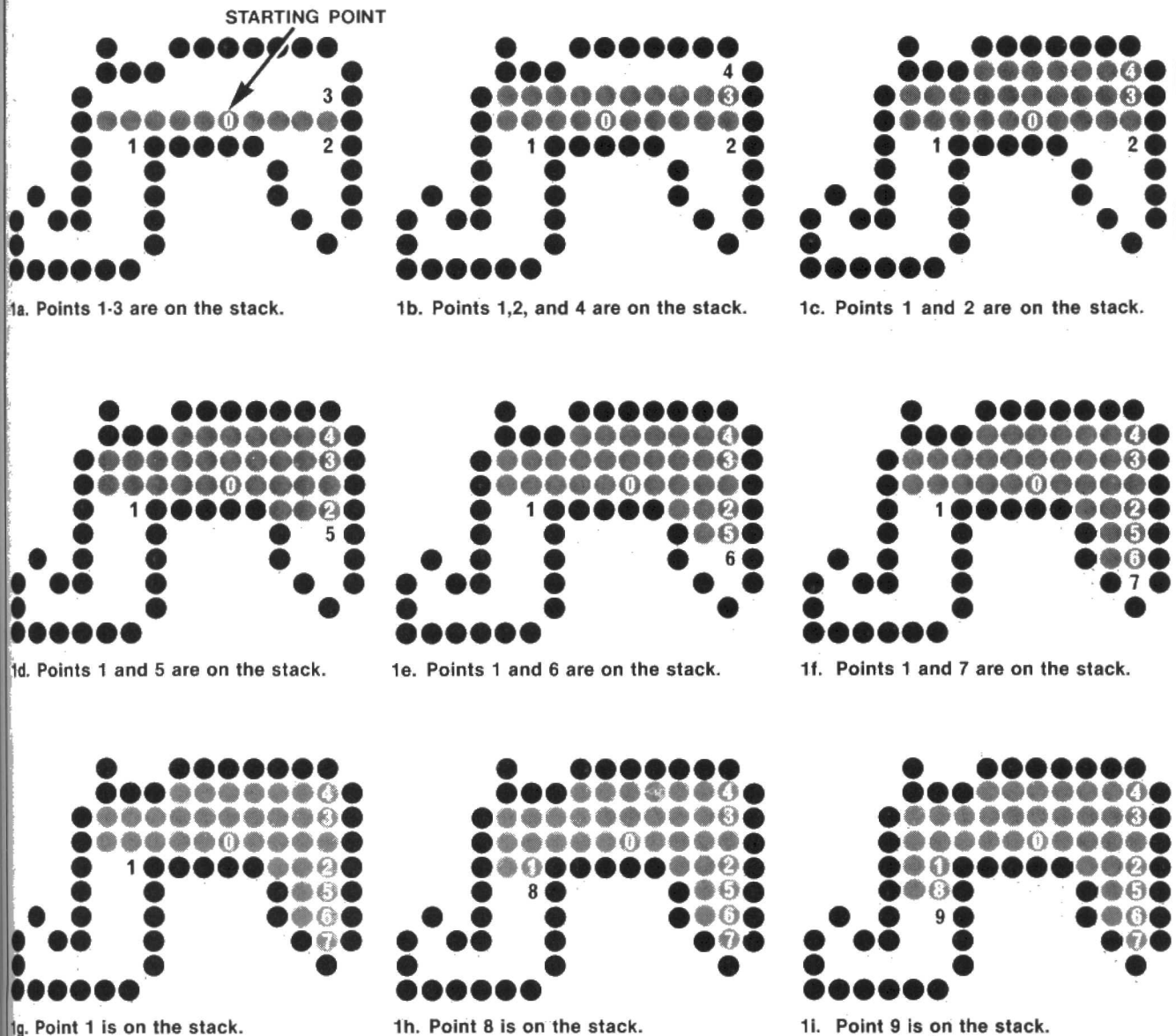


Figure 1. Area filling by runs showing a typically awkward shape.

last two positions of the graphics cursor in user co-ordinates as four 16-bit binary values in an eight-byte control block that may be located anywhere in user RAM. Details of the layout of this block are given in table 3. The routine is entered at &FF1 and is restored through &020C.

Listing 2 contains a procedure called Proclocate, which may be used in other programs: a compressed version of it is used in listing 4. The procedure returns the co-ordinates as four integer variables, x1%, x2%, y1% and Y2%. The values of x1% and y1% correspond to the X and Y co-ordinates of the last position of the graphics cursor: x2% and y2% do the same for the last-but-one position. When used with either of the two new groups of Plot commands, x1%,y1% will be the co-ordinates of the right-most of the two points set, and x2%,y2% will be the left-most point.

The values returned by Osword 13 will always be rounded down to a multiple of the number of logical points, in user co-ordinates, in a pixel. They are stored within the operating-system RAM as addresses in terms of pixels, and are only converted back to user co-ordinates when requested by Osword 13. The conversion includes any resetting of the graphics origin that may have been performed by a VDU 29 call, and the returned values will always describe the pixel in which the originally plotted point lay.

Listing 2 simply draws a line to a random position on the screen and prints the positions read back by Osword 13 on each time round the main loop. Of course, the last-x and last-y values at any time will shift to being the last-but-one-x and last-but-one-y values on the next loop of the program.

When you want to fill an area of the

screen you are confronted with an area of pixels in a background colour, surrounded by a border of pixels in some other colour, possibly more than one. This border may be only one pixel wide, but may be more. It can be very irregular, with lumps forming a significant portion of the area to be filled. Figure 1 shows a typically awkward shape.

A filling algorithm must examine the pixels surrounding the filled area and fill in those that are in the background colour. The simplest technique for doing this is shown in listing 3. It embodies a simple recursive algorithm which will fill any area provided that all the pixels belonging to it share at least one edge with another member.

Such an area is known as a four-connected area, as opposed to the eight-connected type of area where two pixels

(continued on next page)

Colouring blocks

(continued from previous page)

are considered to be part of the same area if they simply touch at a corner. Figure 2 shows examples of both kinds of area. The line-drawing algorithms used on the BBC Micro make four connected areas and would allow an algorithm intended to fill eight-connected areas to leak through any diagonal line. I will confine this article to four-connected areas, but you can find more about eight-connected areas in Foley and Van Dam's quite excellent book *Fundamentals of Interactive Computer Graphics*.

When you try out program 3 on any but the smallest target areas the recursive

Table 1. Effect of Plot options 72 to 79.

- 72 — X and Y are relative co-ordinates; no line is drawn.
- 73 — X and Y are relative co-ordinates; a line is drawn between the two points in the current graphics foreground colour and action
- 74 — X and Y are relative co-ordinates; a line is drawn in the logical inverse of the current foreground colour
- 75 — X and Y are relative co-ordinates; a line is drawn in the current graphics background colour
- 76 — X and Y are absolute co-ordinates; no line is drawn
- 77 — X and Y are absolute co-ordinates; a line is drawn in the current foreground colour and action
- 78 — X and Y are absolute co-ordinates; a line is drawn in the inverse of the current foreground colour.
- 79 — X and Y are absolute co-ordinates; a line is drawn in the current background colour.

Table 2. Effect of Plot options 88 to 95.

- 88 — X and Y are relative co-ordinates; no line is drawn
- 89 — X and Y are relative co-ordinates; a line is drawn in the current foreground graphics colour and action
- 90 — X and Y are relative co-ordinates; a line is drawn in the inverse of the current foreground colour
- 91 — X and Y are relative co-ordinates; a line is drawn in the current background colour
- 92 — X and Y are absolute co-ordinates; no line is drawn
- 93 — X and Y are absolute co-ordinates; a line is drawn in the current foreground colour and action
- 94 — X and Y are absolute co-ordinates; a line is drawn in the inverse of the current foreground colour
- 95 — X and Y are absolute co-ordinates; a line is drawn in the current background colour

Table 3. Control block for Osword 13

00,01 — x co-ordinate of last-but-one position of the graphics cursor, x2%
 02,03 — y co-ordinate of last-but-one position of the graphics cursor, y2%
 04,05 — x co-ordinate of the last position of the graphics cursor, x1%
 06,08 — y co-ordinate of the last position of the graphics cursor, y1%
 To locate 00 in the central block the routine is entered at &FF1 and is vectored through &0206.

algorithm runs out of memory even with the minimal memory used by the mode 4 graphics screen and the trivially short main program. For small, complex areas this program can be useful, but it is not adequate for large areas with the amount of memory available in a non-professional system. The recursive procedure PROCec flood is called about $4*n + 2*m$ times where n is the total number of pixels within the area and m is the number of pixels within the border of the shape. Each call requires memory to hold the two parameters and the return address, so memory runs out pretty fast.

What is needed is an algorithm that is rather logically complex but uses no recursion at all. The new routines are the

fundamental operations for using this algorithm, and it is surprising that Acorn did not finish the job and add a full area-filling routine. Perhaps it will be included in the Graphics Extension ROM, when it appears.

A workable version can still be implemented in Basic, and appears in Program 4. It runs in horizontal rows of pixels within the area to be filled, ending at a boundary of the area at each end. While it is not recursive, the procedure uses a stack on which the positions of the right-hand ends of all earlier unfilled runs are stored. The program uses separate stacks for the X and Y co-ordinates for the sake of simplicity.

When a run is filled, the space above and below it is searched for unfilled runs and stacks the co-ordinates of the right-hand ends of any unfilled runs it finds. The search uses both the new groups of Plot commands. The co-ordinates of the points to be stacked are found using the new Osword call, and the routine ends when the stack is empty.

Figure 1 illustrates how the search works. Relative co-ordinate Plot calls are used with the variable dx% because neither group of calls can move the graphics cursor off the area of colour that it started in, but only up to a boundary

Listing 2.

```

10 REM Program 2
20 REM Demonstrates use of OSWORD 13
30 REM by DRAWING to random locations
40 REM on the screen, and then print-
50 REM ing them out.
60
70 MODE4
71
72 REM Allocate parameter space for
73 REM OSWORD call.
74
75 DIM cords 7
76
80
90 REM Set text window
95
100 VDU 28, 0, 4, 39, 0
110
120 REPEAT
121
122 REM Main loop of program
130
140 DRAW RND(1279),RND(864)
145
150 PROClocate
155
160 PRINT "Last x" "x1X
170 PRINT "Last y" "y1X
180 PRINT "Last-but-one x" "x2X
190 PRINT "Last-but-one y" "y2X
195
200 PRINT "Press SPACE to continue";
201
202 REPEAT
203 UNTIL INKEY(-99)
205
210 UNTIL FALSE
220
5000 END
10000 DEFPROClocate
10006
10010 REM Reads last position of the
10020 REM graphics cursor into x1X,y1X
10030 REM and the last-but-one position
10040 REM into x2X,y2X.
10050
10060 AX=13
10070 X1=cords MOD 256
10080 Y1=cords DIV 256
10090
10100 CALL&FF1
10110
10120 xyoX=cords!0
10130 xynX=cords!4
10140
10150 x1X = xynX MOD &10000
10160 y1X = xynX DIV &10000
10170 x2X = xyoX MOD &10000
10180 y2X = xyoX DIV &10000
10190
10200 ENDPROC

```

Listing 3.

```

10 REM Program 3
20 REM Demonstration of simple
30 REM recursive area filling.
40 REM
50 REM J.G.Dallman, June 1983.
60 REM
70 MODE4
80 REM
90 REM Draw a small, simple shape
100 REM to fill.
110
120 MOVE 600,500
130 DRAW 650,500
140 DRAW 650,350
150 DRAW 600,350
160 DRAW 600,500
170
180 REM And fill it in..
190
200 PROCrec_flood(625,525)
210
220 REM Wait for user
230
240 PRINT "Press SPACE to continue"
250
260 REPEAT
270 UNTIL INKEY(-99)
280
290 REM Draw a larger shape
300
310 CLS
320
330 MOVE 300,300
340 DRAW 700,300
350 DRAW 500,700
360 DRAW 300,300
370
380 REM And fill it in - or try...
390
400 PROCrec_flood(500,500)
410
420 END
430 DEF PROCrec_flood(x,y)
440
450 REM Have we reached the edge of
460 REM the shape ?
470
480 IF POINT(x,y) > 0 THEN GOTO 600
490
500 REM We haven't, so fill it in
510
520 PLOT 67,x,y
530
540 REM And try the points around it
550
560 PROCrec_flood(x-4,y)
570 PROCrec_flood(x+4,y)
580 PROCrec_flood(x,y-4)
590 PROCrec_flood(x,y+4)
600 ENDPROC

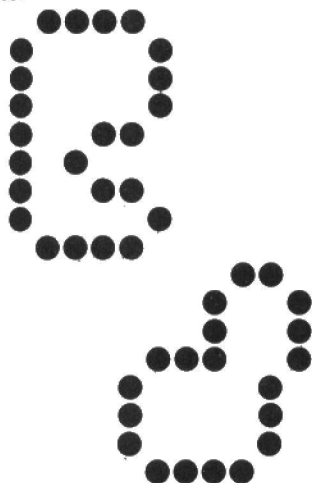
```

The variable `dx%` is set to the horizontal size of a pixel in the current screen mode by `FNhorstep` so that the move over the boundary can be performed reliably.

Very little stack space is used by this technique, except for areas that branch out into many small ones. A machine-code version should therefore be quite practical, even given the restricted stack space available on the Micro's 6502 microprocessor. The new Plot routines are not very fast, and an instantaneous fill seems impossible. Taking Rem lines out of the Basic version and using multi-statement lines can speed it up about 30 percent, although that is still rather slow for use within applications programs.

It is possible to extend the filling technique to fill areas of foreground colour, and to change the colour of ready-filled areas. More details can be found in A R Smith's article, Tint Fill, in *Computer Graphics*, August 1979. Acorn's routines in their current form may not allow all the possible extensions. In that case disassembly of the relevant sections of the operating-system ROM should give a good idea of how to write additional routines along the same lines.

Figure 2. Four- and eight-connected area.



2a. Four-connected areas.



2b. Eight-connected areas.

Listing 4.

```

10 REM Program 4 -
20 REM Demonstration of non-recurs-
30 REM ive area filling.
40 REM
50 REM J.G.Dallan, June 1983.
60 REM
70
80 MODE4
90
100 REM Dimension parameter block for
110 REM PROClocate.
120
130 DIM cords 7
140
150 REM Dimension arrays for software
160 REM stacks.
170
180 DIM sx(128),sy(128)
190
200 SPMAX = 0
210
220
230 REM Draw a large, complex shape
240 REM to fill.
250
260 MOVE 200,200
270 DRAW 250,450
280 DRAW 500,100
290 DRAW 1000,270
300 DRAW 1100,0
310 DRAW 1200,800
320 DRAW 1100,250
330 DRAW 500,500
340 DRAW 400,400
350 DRAW 200,500
360 DRAW 200,200
370
380
390 REM Fill in the shape
400
410 PROCflood(450,300,1)
420
430
440 REM Print maxium value of software
450 REM stack pointer.
460
470 PRINT "SPMAX = "; SPMAX
480
490 END
500
510
520 DEF PROCflood(X,Y,tint)
530
540 REM Initialise variables.
550
560 stackptr% = 81dx% = FNhorstep
570 stacktop% = -1stacktop% = -1
580
590 REM Set colour for filling.
600
610 GCOL 0,tint
620
630 REM Push starting co-ordinates
640 REM for first time round loop.
650
660 PROCpush(X,Y)
670
680
690 REM Main loop of procedure
700
710 REPEAT
720
730 REM Pop starting co-ordinates
740 REM off software stacks.
750
760 PROCpop
770
780 REM Fill across this y-co-ordinate.
790
800 PLOT77,stacktop%,stacktop%
810
820 REM Locate points and set up the
830 REM **base% variables.
840
850 PROClocate
860
870 xibase% = x2%yibase% = y2%
880 xrbase% = x1%yrbase% = y1%
890
900
910 REM screen & try to fill the line
920 REM above the current one.
930
940 IF POINT(xibase%,yibase%-4)<>-1 THEN
950 PROCsearch(-4)
960 IF POINT(xibase%,yibase%+4)<>-1 THEN
970 PROCsearch(4)
980
990 UNTIL stackptr% = 0
1000
1010 REM We've run out of lines to fill
1020
1030 ENDPROC
1040
1050 DEF PROCsearch(dy%)
1060
1070 REM Searches for unfilled runs on
1080 REM the line above or below the
1090 REM current one.
1100
1110 LOCAL x%,y%
1120
1130 REM We have started above/below a
1140 REM point that can be lit or not -
1150 REM this handles that case.
1160
1170 IF POINT(xibase%,yibase%+dy%)=0 THEN
1180 PLOT76,xibase%,yibase%+dy% ELSE PLOT79
1190 2,xibase%,yibase%+dy%PLOT72,dx%,0
1200
1210
1220 REM Locate the point (the right
1230 REM end of a run) found above.
1240
1250 PROClocate
1260 flag% = FALSE
1270 REPEAT
1280 REM Search rightward for right ends
1290 REM of runs and push them onto the
1300 REM software stacks.
1310 IF x1% > xrbase% THEN flag% = TRUE:
1320 GOTO 1340
1330 PROCpush(x1%,y1%)
1340 PLOT72,x1%+dx%,y1%
1350 PLOT72,dx%,0
1360 PROClocate
1370 UNTIL flag%
1380
1390 REM The search has gone beyond the
1400 REM right end of the previous run.
1410
1420 REM Check if the run we're working
1430 REM on extends beyond the base run
1440 REM to the right - if so find it's
1450 REM right end and push that.
1460 IF POINT(xrbase%,yrbase%+dy%)=0 THEN
1470 PLOT76,xrbase%,yrbase%+dy%PROClocate
1480 iPROCpush(x1%,y1%)
1490
1500 ENDPROC
1510
1520 DEF PROCpush(x%,y%)
1530
1540 REM Put x% and y% onto the soft-
1550 REM ware stacks and into stacktop%
1560 REM and stacktop%+1.
1570
1580 REM Check we haven't stacked these
1590 REM points already.
1600 IF x% = stacktop% AND y% = stacktop%+1
1610 THEN GOTO 1660
1620
1630 REM Perform stacking
1640
1650 sx(stackptr%) = x%sy(stackptr%) = y%
1660 stacktop% = x%
1670 stacktop% = y%
1680 stackptr% = stackptr%+1
1690
1700 REM Check value of software stack
1710 REM pointer.
1720 IF stackptr% > SPMAX THEN SPMAX = stackptr%
1730 ENDPROC
1740
1750 DEF PROCpop
1760
1770 REM Pop software stacks and set
1780 REM new values of stacktop% and
1790 REM stackptr%.
1800 stackptr% = stackptr%-1
1810 stacktop% = sx(stackptr%)
1820 stacktop% = sy(stackptr%)
1830 ENDPROC
1840
1850 DEF FNhorstep
1860
1870 AX = 13:K2 = cords MOD 256
1880 YK = cords DIV 256
1890 CALL &FFF1
1900 xyo = cords!0
1910 xyn = cords!4
1920 x1% = xyn MOD &10000:y1% = xyn DIV &10000
1930 x2% = xyo MOD &10000:y2% = xyo DIV &10000
1940 ENDPROC
1950
1960 DEF FNhorstep
1970
1980 REM Returns horizontal size of
1990 REM pixel in current mode.
2000
2010 AX = 135
2020 mode% = USR(&FFF4)
2030 mode% = mode% AND &FFFFFF
2040 mode% = mode% DIV &10000
2050 IF mode% = 3 OR mode% = 5 THEN PRINT "Wrong
2060 mode, dummy!!":STOP
2070 IF mode% = 0 THEN = 2 ELSE IF (mode% = 1 OR
2080 mode% = 4) THEN = 4 ELSE = 8

```

References

- Beebug Newsletter, *Acorn News*, Volume 1, Number 6, October 1982.
- Fundamentals of Interactive Computer Graphics* by Foley and Van Dam, Systems Programming series, 1982. Published by Addison-Wesley.
- Tint Fill by A R Smith in *Computer Graphics*, August 1979.

FOR SOME TIME the subject of image analysis has interested engineers and computer scientists. The ability to put a graphical representation of a real object into a computer and manipulate it has found countless applications from pattern recognition to CAD. Until recently most of the available vision hardware was based around either mainframes or specially designed processors; it did not have the general applications which would allow high-volume sales over which to spread development costs.

The microcomputer has a wide range of applications from office administration to process control and is proportionally lower in cost. Rather than using external hardware such as CPUs and RAM to capture the image the micro's own hardware is used as a frame store. The data becomes much more accessible to the user for the purpose of image processing.

Solid-state cameras provide the computer with digital data representing a map of pixels, which make up the image falling on the sensors of the camera's solid-state array.

Due to the low level of production solid-state cameras can be quite expensive. This is changing and solid-state sensors are being applied to a range of consumer products. Some solid-state cameras use optic RAM rather than a sensor. This enables an image to be fed straight into RAM, and the sensor itself to be read in the same way as a frame store. It provides a very rapid, low-cost method of obtaining a binary image.

The alternative to the solid-state camera is the Vidicon television camera which provides an analogue signal, which is then digitised. Unless expensive A to D

Images of digits

Peter Kruger and Stephen Cronk of Digithurst Ltd explore the potential for high-resolution vision systems.

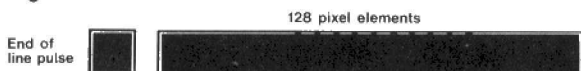
converters are used this method is slower than the solid-state camera, but it does have the advantages of being both low cost and giving a grey-scale output. In general terms, where the object being analysed is slow moving and a grey-scale image is required a television camera system can be used. If a rapid access time is required and a binary image is sufficient, then a solid-state system should be used.

To carry out analysis the image data can either be sorted in external RAM, in the camera or frame store, or in the microcomputer itself. Advantages and disadvantages are present in both systems. The cost of external memory and the extra processing required is high, but may be necessary if the image analysis required is complex and takes up a large amount of

RAM. If the microcomputer is large enough to hold the image and the software required, then the data becomes much more accessible to the user and the cost of additional hardware falls.

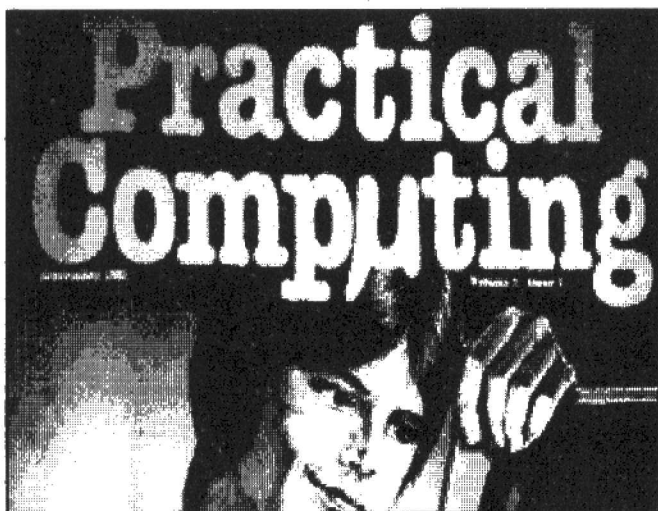
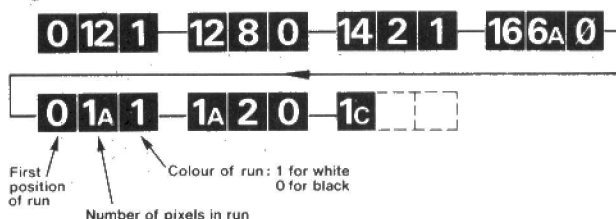
Once the image has been captured and stored it may be desirable to display it. The image may be shown as a binary or a grey scale. If a binary display is required the data must be scanned and each pixel compared to a preset threshold before deciding to display them or not. To simulate grey scale a group of screen pixels are used to represent each camera pixel. The thresholding technique is used for each pixel within the group and an image is displayed not dissimilar to newsprint. A contrast value can be used to set the threshold values for the pseudo grey-scale

Figure 1.



The data structures hold data in unprocessed or processed forms. Figure 1 shows unprocessed data, which is held in RAM. Figure 2 shows processed data which is unprocessed data that has been encoded.

Figure 2.



A grey-scale image.



A binary image.

display. Both routines can be written using the computer's point-plot routine making the software transportable between different micros.

Data may take one of two forms when it is read into the computer. It may either be binary image data and be bit mapped or it may be grey-scale data and be byte mapped, each byte having a value representing the brightness of the respective camera pixel. At this point it will be necessary to clean up the image. High-resolution pictures gained with a video camera which has random interlace must have the effect of the interlace removed, which is achieved using recursive processing.

This is a relatively simple yet effective way of reducing noise or any form of sporadic interference on a digitised television picture. A number of frames are captured, each being averaged with the previous using the algorithm:

$$\text{NEW PIXEL} = (\text{OLD PIXEL} + \text{INCOMING PIXEL})/2$$

The random nature of the interference means that over a number of frames the unwanted noise will tend to cancel out. Increasing the number of frames captured and averaged in this way improves the final

result but also increases the time taken to reach that result. It is usually found that acceptable results are achieved after the first three or four frames; after that the small improvement in picture quality is minimal compared to the extra time needed.

A slightly more advanced version of the technique which leads to more flexible filtering allows the user to define the proportion of the incoming image that is mixed with the previous image, using the algorithm:

$$\text{NEW PIXEL} = K * \text{OLD PIXEL} + (1 - K) * \text{INCOMING PIXEL}$$

where K is a user-specified constant weighting the new image.

Recursive processing techniques can also be used to intensify a low-level video signal such as one that is shot out of doors at night. Each captured frame is summed with the previous frame so that over a period of about 10 frames, depending on the light level, a clear image can be seen. The process requires the image to be stored in 16-bit words as it is quite likely that the values obtained may be greater than 255.

To remove any electrical noise appearing as individual pixels, or marks and small objects which appear as single pixels and

therefore cannot be verified at the current resolution, the image data is cleaned. The cleaning consists of examining pixels in groups of three and eliminating any pixels whose neighbours differ radically in intensity.

Image compaction techniques can be used to reduce the size of the image-data file to speed up data access during future processing. One method of data compaction is run-length encoding which reduces the memory required to hold an image by up to 16 times. Each pixel is examined and compared with the current threshold value. The next pixel in the current raster is also compared to the threshold value, and a run of pixels of the same thresholded value is built up. Each run is stored in a three-byte record, the first byte giving the start point of the run, the second byte the number of pixels in the run, and the third the colour of the run.

The amount of grey-scale and processed data which can be held at any one time will depend on the memory size of the computer being used. For example, a 256K Sirius will hold a 256 by 256 grey-scale image occupying 64K as well as at least one processed image at any one time. This allows the image to be processed at various thresholds without disturbing the original data.

Object/pattern recognition can be undertaken either by using values of area and perimeter or by examining the grey-scale data in greater depth. By using edge-detection methods it is possible to build up a series of co-ordinates which can be passed to a CAD software package.

With the introduction of more powerful microcomputers and with greater speed and memory mapping capabilities there is a greater potential for higher-resolution vision systems. To anyone who has spent hours inputting graphics into their programs the applications of low-cost vision systems is obvious. Images can be used as backgrounds for games programs or computer-aided learning software with images being read in and reduced to line drawings in a matter of seconds.

Solid-state cameras provide the computer with digital data.



Grey-scale image with edge detection on one section.



Inverted image with edge detection on one section.

Death to the machine

Mandy's eyelashes flickered open, slashing the teardrop that welled in the corner of her eye. Memories of the night's storms overwhelmed her. She drew back, hating the world that had invaded her sleep and ruthlessly dragged her back to reality.

A dog barked in the avenue below. A blackbird's vivacious song rang out. Dazzling sunshine illuminated the curtains and showered the room in gold dust.

Henry sensed her waking, and he remembered the way she had quaked when the rolls of thunder and lightning had terrorised the night. Gently he touched her skin. To reassure her, he told himself.

They lay together side by side, aware of each other but not speaking, choosing instead the intimate silence of lovers. From the corner of her eye she watched him, his body bronzed in the mist of sunlight. She was afraid to move, to speak, afraid of destroying the moment.

At last he rose, silhouetted against the curtains, innocent of his own nakedness and his partner's idolatrous gaze. She studied him jealously — his flawless skin, his broad shoulders and pronounced muscles, his grace. He moved like a panther.

"Good morning Amanda," he said with a sparkling smile.

"Morning Henry," she whispered.

He carried her across the room and deposited her in the wheelchair that stood dormant in the corner. He felt no revulsion at the ugliness of her wasted limbs. Sympathy never crossed his mind.

Mandy watched him carefully. She hesitated nervously, then at last plucked up courage to speak.

"Last night . . . Henry. Thank you . . . I know it broke all the rules. But I needed you."

Henry nodded a silent acknowledgement.

"Lights. Curtains." He gave the command as he left the room. The lights dimmed themselves and the curtains drew back, engulfing the room in light.

Mandy tucked into her breakfast with relish, scooping up the pieces of bacon with the fork in her right hand. The left was draped uselessly in her lap.

"Television 1," she ordered, and the

television obligingly flickered into life. The Breakfast O'Clock News held her attention as she ate.

An explosion. The crashing and splintering of glass and wood. People running, shouting, throwing, hating. The crack of gunfire. Police armed to the teeth, charging. Panic. People screaming.

Mandy was sickened, but the screen compelled her to watch, holding her eyes the way a swaying cobra hypnotises its prey.

"A spokesman for the company, Robo-of-America, said that 10 robots were

by Andrew Walker

completely destroyed and several others had been severely damaged, putting the cost at 13 million dollars, 37 rioters were reported killed."

The robot newsreader spoke in cold tones, reading the idiot-tape that ran through his wrist. "The President attacked the left-wing militants who, it was said, were using people in a political game. By telling people that big business was replacing humans with robots, the communists were feeding on the fears of the working class for their own subversive ends. She added that we must all make sacrifices."

On the screen a robot was being dismembered by the rioters, while another was catapulted from a third floor window. Henry shuddered.

But Mandy was bored, numbed to the violence by its day-by-day repetition.

"Shopping," Mandy commanded.

The television picture blinked out and a menu appeared, cursor flashing. Her fingers played deftly on the console installed on the right arm of her chair. She looked for things they were running out of — food, polish, toilet rolls. She compared prices and ordered items. She picked the colours that took her fancy and went window shopping for the latest fashions. New screens continually sprang into view, choices were made at leisure.

"What do you want to do today?"

Henry fired the question into the air as he worked, not looking at Mandy.

"Take me home, please. You know I want to go."

"This is your home."

"My real home," she pleaded.

Henry paused.

"You know what I think about going there. I don't like it. It's dangerous — full of thugs and hookers."

"And it's not your home — not any more. It's not the quiet suburb of your childhood. Wipe those memories away for your own sake."

"Take me," Mandy persisted. "Please."

Henry bit his lip, but replied reluctantly. "All right. You know I can never say no to you."

The Hill was home. Tree-lined boulevards, pipe-smoking artists lazing on street corners, discussing Picasso and extolling the beauty of the girls as they walked by. Sunny days, and families taking the air in their Sunday best, nodding to passing acquaintances.

It was all long gone. Buildings rotted and neglected streets flowed with garbage. The Hill groaned under man's physical graffiti.

Henry sensed the eyes on them, strangers in a strange land, anachronisms, belonging even to a different species. Faces hid behind curtains that flapped in open windows, through which the shadows of the rooms within seemed to give each building an aura of dirt and doom.

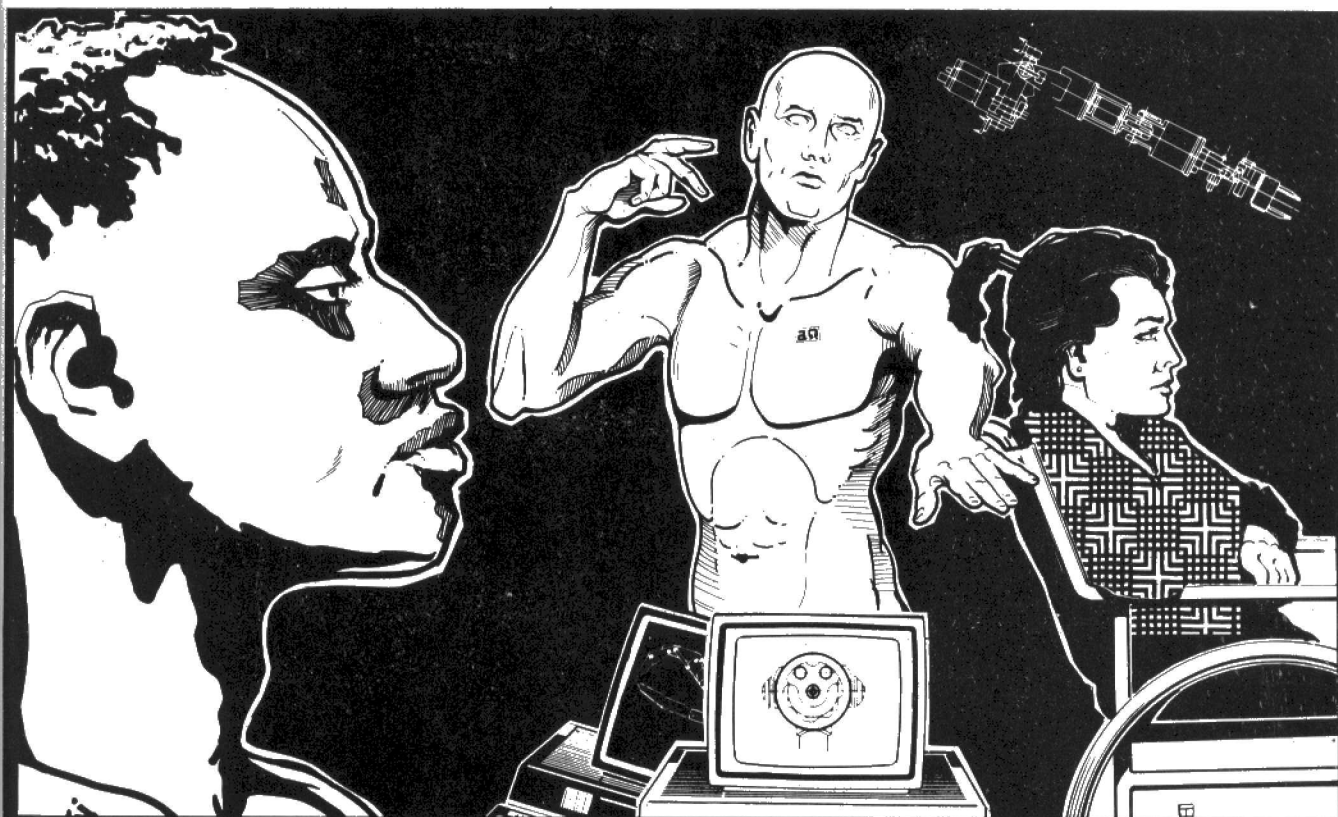
Mandy saw her childhood. The family and the home. The breaking of morning across the rose-tinted garden. Her friends. The birds that spoke to her. The way she would run through the dew-laden grass, chasing or being chased, laughing and shrieking with joy.

"Satisfied?" Henry asked malevolently. "See how ugly this place is. We shouldn't have come."

But Mandy was far away.

"Smash the bastard!"

The mob fought hard for the right to kick and punch its victim. People ran, jostled, writhed like maggots, a seething mass of frustrated, angry bodies. They vented their hatred, emptied their sack of fears, and found relief in aggression.



Moving down the street, heedless of passers-by, the amoeba broke and reformed, then broke again, and finally raced ahead, exhilarated by pain and violence. Each man was caught by the mood of blind rage, joining in the baiting and killing of guilty and innocent.

They tired, but as they dropped their places were taken by others, ever more eager, ever fresh and hungry, charged by the electricity of the moment. They grabbed a drone as he hauled a crate 10 times his size, descending on him like vampires. He struggled briefly before drowning in a sea of people. Cheering people, revelling in destruction, throwing parts of the drone hither and thither, people turned mad on a balmy Sunday afternoon.

"This one won't steal a man's job again. Get out of the way — let me kick it. Death to the machine!"

Henry shivered as he looked at the grieving, stricken torso. A stranger staggered along the street from the direction of the mob. His head hung low, but as he neared Henry he looked up with glazed eyes. He was pale as death.

"They ... they ripped him apart." He trembled with each word. "A man. They did it to a man."

Henry turned, leaving the stranger retching in the gutter. He pushed Mandy along, glowering at the back of her head.

"Damn you!" he thought. "I told you."

And his circuits wondered at this new emotion.

"There's another!" voices cried.

Fingers darted in a million

directions. The voices grew in number. Factions split, intoxicated by violence, clashing over their prey like packs of hyenas.

"There's one!"

The cry pealed out like church bells, across the streets and through the alleys, to shoppers, to people leaning from windows with ghoulish curiosity, to a mongrel that paused briefly as it chewed a robotic leg, to a gang prising open a Space Invaders machine. To a new victim. It tolled the death knell.

A strong, steady finger pointed. The finger of the hard man. A compelling finger that urged the eye to follow its line to the guilty man who stood there dumbfounded, still holding on to the wheelchair in which his prisoner sat, a poor, weak-looking human girl.

The cry tolled out for Henry.

He broke into a run. With a wave of his arm the hard man commanded the crowd of onlookers to give chase. But they were on their way already, stirred by propaganda, feverishly wanting to free the girl from the inhuman monster.

The wheelchair careered madly as Henry galloped, hounded by the baying hunters. He swung round corner after corner, in and out of alleyways, trying vainly to shake them off. The mob struggled, a cursing, stumbling rabble. But always he sensed the hard man hard on his heels.

"Leave me Henry. Run."

Mandy's pleas grew as the wheelchair rocked and rolled and Henry's silence lengthened.

His circuits burst with energy,

transceiving messages. Logic chips warned him of the danger to Mandy, how she would be mistaken for a robot, how she would be mutilated, how the humans would kill her from instinct alone. Chips of intuition calculated probabilities — the chance of her neck being broken if they kept the present speed, her chances of survival if he left her.

"We are 74 percent certain that a savage band of humans would not harm a cripple girl," they said. "Save yourself," cried out self-preservation units.

But wise old legal chips read the rules: "You cannot leave her to a doubtful fate; preservation of the machine is secondary. Self-sacrifice is an honourable death."

"Leave me," Mandy added.

"Can't," Henry stammered, finally, through gritted teeth.

"Run Henry, run," Mandy pleaded tearfully.

"Run rabbit, run," bellowed the hard man, like a blast from the farmer's gun.

The packs closed in, howling triumphantly as their prey froze, each eager to cheat the other of its kill. And then the hard man stood face to face with Henry.

Henry's brain tore into itself. Logic clashed with law. Morality proclaimed. Self-preservation cried it down. "Kill him — you can't kill a human — it's your only chance — his life or hers and yours — Primary Robotic Law states ..."

Anonymous circuits assented or dissented in a clamour of a billion electronic pulses. He clutched his head. He

(continued on next page)

Death to the machine

(continued from previous page)

argued and proposed solutions and then dashed them all against the wall of robotic law.

The hard man smiled knowingly. With a crayon he began to daub Henry's forehead in blue: "666". The number of the robot, reckoned the propagandists.

Henry seethed with rage. He let go a bellow, like a trapped animal caught without hope, a bellow that turned to a scream of anger and shame. A scream that became a flash of insanity.

A clenched fist lashed out, breaking the invisible barrier: for an instant Henry became human. The hard man's jaw cracked behind the punch, and like a man possessed, his whole body convulsed in a wild frenzy.

Henry stepped back aghast, shocked by his violence, disowning the fist that had offended, and repulsed by the spectacle and nauseating fumes before him. A flood of integrated circuits, cogs and lengths of

wire poured from the hard man's mouth. His head erupted, shattering into a billion slivers of silicon.

Suddenly all was quiet, his death throes exhausted, his headless corpse frozen upright, erect for eternity. The stupor that hung over the scene held everyone as Henry edged away. No one followed.

"He was just a tool," Henry insisted. "Like the video games, like the bus driver. Like me."

"I don't understand," Mandy repeated the statement for the umpteenth time, chafing Henry's nerves like sandpaper on an open wound.

Birds were singing in the trees. The sun shone from a clear sky. Towering over her stood the city, a warm paternal giant whose strength comforted her, a city of glass skyscrapers, hygiene and automation, where there was light and no shadows, where the people thought of peace and beauty.

Henry knelt in front of her as she wiped the blue scrawl from his forehead, fussing maternally. She avoided his eyes. "He was a robot — all your militants use them," Henry explained. They're programmed to seek out other robots, stir up feeling against them, incite riots: a Judas. Built to slaughter his own."

"It's always the same," he went on. "Cars, television. You invent these things, make the world a better place. Then what? You're lost. You take them for granted,

ignore them, distort them, manipulate them, turn them to evil and then learn to fear and hate them.

"You wonder at your own power: you've made something in your own image — you are God. But you're afraid — it's better than you. So you tear it down. Seek it out, cut it out like a cancer."

Mandy was pinned to her chair by the venom in Henry's tongue and the crazed ravings of his mind. She trembled in fear of the stranger before her.

Circuits raged in new-found freedom, or cowered in seclusion, screaming their terror of this brave new world.

"Closedown."

The command froze Henry in mid-sentence. His eyes closed. He was suspended in time.

"Re-initialise," God commanded.

Henry flickered into life, and a warm, comforting smile spread on his lips.

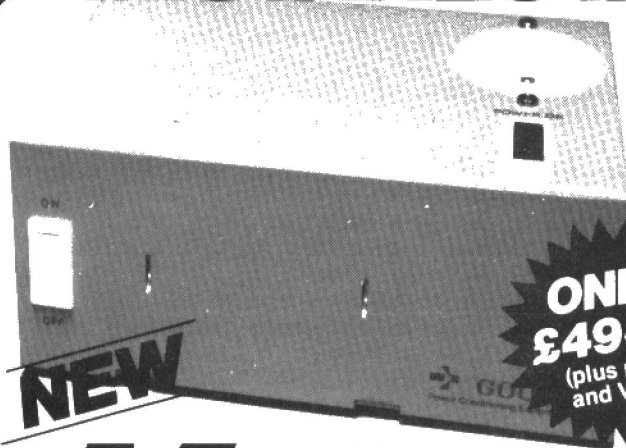
"Isn't it a lovely day, my love?" He spoke in a sickly sweet voice that chilled her spine. "Let's go to the park."

The crisis was over, but a malevolence crept into his face, a malevolence that even God did not detect. It distorted his features and set a darkness in his eyes as it spread to his powerful arms.

Below them the avenue stretched down to the wharf, where streetside cafes vibrated with life, and the sun smiled on the rich, beautiful people. Henry began to loosen his grip.

NOW!

PROTECT YOUR COMPUTER FROM MISTAKES IN THE MAINS



ONLY £49.95
(plus p&p and VAT)

Most unexplained computer errors aren't the fault of the computer or its user, but can be traced to a poor mains supply.

Electrical noise from sources like fluorescent lighting, electric kettles, CB radios and radar can play havoc with data. What's more, the voltage fluctuations from lightning and heavy electrical appliances may even result in permanent damage to memory units.

Now Mainstay stops all that. It literally filters and regulates the voltage going into your computer, as well as supplying microbreak power.

That means total protection for micro and mini-computers from the makers of some of the world's most sophisticated equipment for protecting mainframe computers and systems. Play it safe. Send the coupon today.

Mainstay

MAKE NO MISTAKE!

Gould Power Conversion Division, Rhosymedre, Wrexham, Clwyd, LL14 3YR Wales Telephone (0978) 821000 Telex 61424



PLEASE TICK EITHER

☐ Tell me all the facts about new Mainstay and how I can protect my computer

☐ Please send me p&p and VAT and debit my Access/Visa ac number

Name

Address

Sig

*delete as appropriate

● Circle No. 170

DUPLEX

NOW REQUIRE DISTRIBUTORS WORLD WIDE FOR THEIR OCTET INTERFACES FOR OLIVETTI'S ET 121 ELECTRONIC TYPEWRITER

Duplex has successfully developed two plug-on interfaces for Olivetti's ET 121 electronic typewriter — these are currently being distributed exclusively by British Olivetti themselves in the UK and Eire. The Duplex interfaces enable the ET to be used as a letter-quality printer

and/or terminal for a computer, add an extra 4,000 characters to the ET's one-line memory, enable the ET to

communicate with other Olivetti word processors and typewriters, and allow the ET 121 to be attached to a telex tape punch/reader and tape cassette interface unit to provide infinite external memory. Foreign keyboards and character sets can easily be accommodated.



DUPLEX
PLUG-ON
INTERFACE KSR/MSR

STANDARD
ET 121
TYPEWRITER

GENERAL PURPOSE/ WORD
PROCESSING COMPUTER



ET 121 + OCTET



OCTET T1



COMPUTER



TELEX TAPE
STATION OCTET-PO/PR



THE OCTET 121 — KSR

(Keyboard/send/receive)

- PLUG-ON unit. (30 seconds or less to fit or remove)
- 2000 Char. Print Buffer
- RS232C Specification (TX & RX)
- Three modes of operation (normal/set up/Ksr)
- WHITE SPACE SKIP to increase print throughput up to 45 cps
- Printer Pause/Resume Key
- True KSR (Keyboard/send/receive) mode
- Keyboard SET-UP mode to change operating parameters (non-volatile)
- Xon/Xoff, CDSR/DTR, ETX/ACK handshaking protocols

If you are interested in becoming an exclusive distributor in your country for the Octet interfaces contact D. J. Winder at Duplex. Countries of particular interest include AUSTRALIA, NEW ZEALAND, HONG KONG, CANADA, USA, MIDDLE EAST and most EUROPEAN countries.



THE OCTET 121 — MSR

(Memory/send/receive)

- All KSR features and
- 4000 Char. Working Memory
- Simple to use TEXT-EDITOR
- Five modes of operation (Normal/telex/letter/setup/Ksr)
- Battery-backup & Power-Fail-Safe. (Protects memory contents)
- Allows OCTET series products to be interconnected
- Special FORM-FILLING facility using multiple stop-codes
- Allocate up to 100 special phrases (paras or whole letters)
- Standard ET 121 features retained at all times

DUPLEX
Communications Ltd.

The Interface People

Midlands/North — 2 Leire Lane, Dunton Bassett, Nr. Lutterworth, Leicestershire LE17 5JP. Tel: 0455 209131 or 202150

South — 52 High Street, Stock, Essex CM4 9BW. Tel 0277 841011

● Circle No. 171

A question of significance

With Owen Bishop's Basic programs you can tell chance from choice.

"COFFEE, TEA OR ME?" Ignoring the "Me" of the memoirs of those incorrigible airline stewardesses, Trudy Baker and Rachel Jones, still leaves you with a choice to make: shall it be tea or shall it be coffee? It must be one or the other, but you are not allowed to have both. To put it more technically, they are mutually exclusive choices.

This kind of choice is thrust upon you many times a day. Sometimes you make a response based on a strongly felt preference or a reasoned argument. But often you are indifferent and decide on a whim or by tossing a coin. Your choice might just as well be the result of running:

```
20 IF choice = 1 THEN PRINT "Coffee"  
ELSE PRINT "Tea".
```

Suppose a board of directors is offered tea or coffee and all choose tea. Does it mean that they genuinely prefer tea? Is it worthwhile brewing up coffee next time? A majority of six to none seem a strong one, but can you be sure that the board has a genuine preference for tea?

There is no need to go into the reason for the preference, if there is one: the tea may be superb, the coffee may be like dishwasher or maybe they are just a bunch of cha-wallahs. You just want to know whether they have a genuine preference or made their choice through whim.

Suppose the directors had no strong

reasons for their choices and each decided to run the random-choice program on the firm's mainframe and imbibe accordingly. Any given director is equally likely to select tea or coffee, unless there is a bug in the mainframe or its random-number algorithm. Six different outcomes are possible:

all six choose tea
five choose tea and one chooses coffee
four choose tea and two choose coffee
three choose tea and three choose coffee
two choose tea and four choose coffee
one chooses tea and five choose coffee
all six choose coffee

These seven eventualities are not equally likely to occur. For instance, there is only

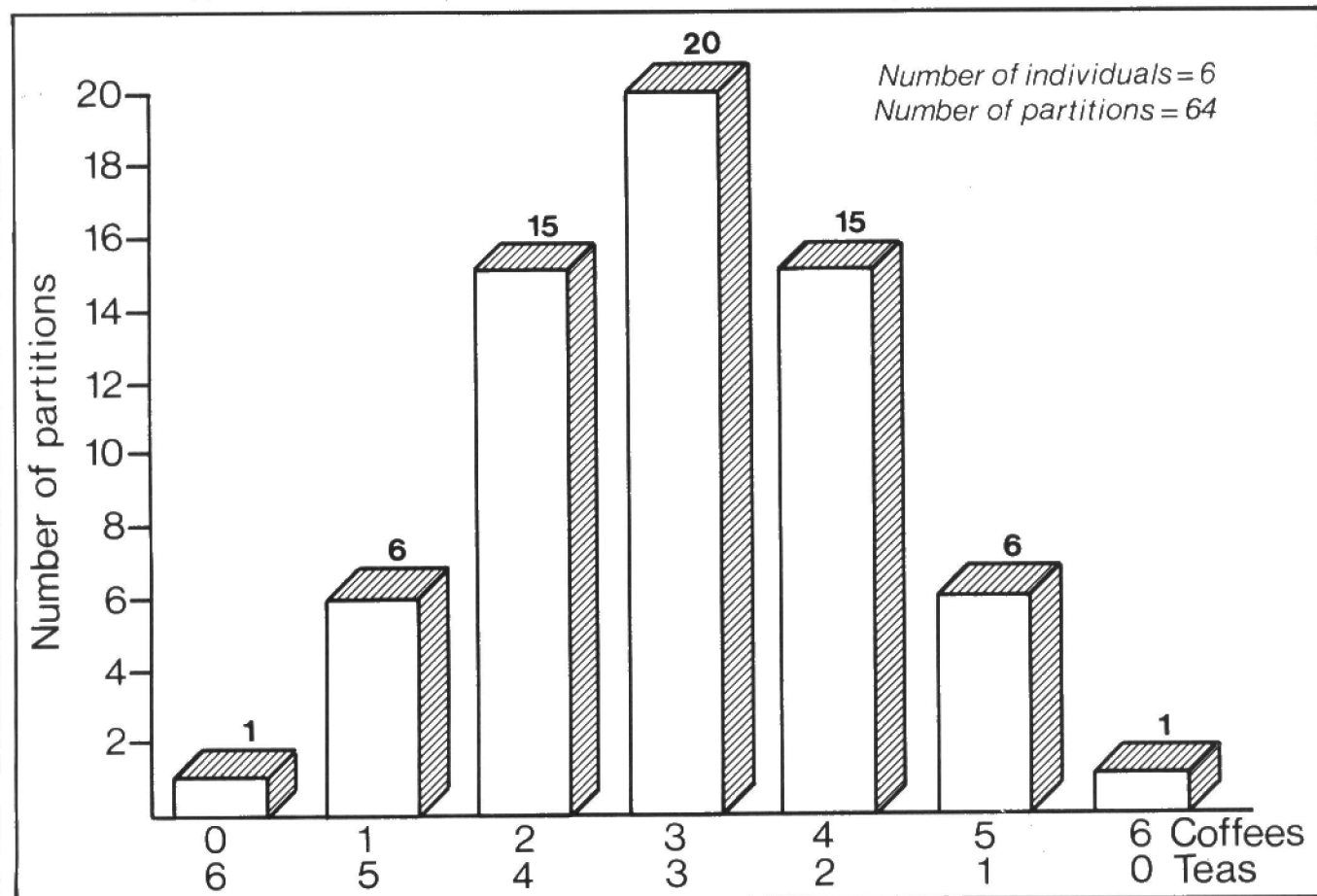


Figure 1. There are 64 ways in which six people can choose between tea or coffee.

one way in which all can choose tea, and only one way in which all can choose coffee. There are six ways — one for each director — in which one chooses coffee while the others have tea, and six ways of serving one tea and five coffees.

For two coffees and four teas you have to work out how many ways there are of picking out the two coffee-drinking directors. It is best done by representing the directors by the initial letter of their first names — they are called Alf, Bert, Connie, Dick, Evie and Fred — and listing them as in table 1 to what the computer tells them to choose.

The groupings in the middle column represent all the possible ways of picking out those who are to have coffee. There is no need to go further, for the arrangements are symmetrical — see figure 1.

There are 15 ways in which four choose coffee, six ways in which five choose coffee and one way in which all choose coffee. The total number of ways in which the directors can be divided into coffee drinkers and tea drinkers is $1 + 6 + 15 + 20 + 15 + 6 + 1$ or 64 ways.

If all choose tea there is only one way out of the 64 ways that this can happen randomly. Their behaviour can be explained in two ways:

- they really do prefer tea
- they are choosing at random, and a one-in-64 event has occurred.

It is safe to infer that their apparent preference for tea is genuine. On the other hand, there still remains the one-in-64 chance that it is a random choice, so unless you are prepared to take that risk of being wrong, it is wise to keep coffee on the menu for future board meetings.

But what if one director had chosen coffee? The calculations show that there are six ways of this happening in a purely random way. You could still believe they prefer tea but there is a much stronger risk that their corporate choice was made at random. There is now a six-in-64 chance, around nine percent, that you are observing random choices and not real decisions.

You could also arrive at the same conclusion if you had just witnessed the directors voting five to one in favour of investing half the company's funds in a new expansion project. With so many factors involved in the market the outcome is virtually unpredictable, and a random choice may be as good as any. With six directors, only a unanimous decision can signify anything to a mere outsider.

The scene now changes to a school biology laboratory where the students are watching six woodlice in an apparatus called a choice-chamber. Incidentally, the woodlice are also called Alf, Bert, Connie, Dick, Evie and Fred. They are being made to choose between going to a specially prepared damp part of the chamber or to a part which is dry.

As it turns out, they all go to the damp part. You have not been able to ask them which they prefer, and before the test was run you did not know which they would prefer. You took care to design the apparatus so that the two sections of the chamber should be equal in all respects other than dampness, and that the woodlice should be put into the chamber at a point where they were equidistant from both.

As with the directors, so also with the woodlice, you must have a unanimous

decision when there are only six choosers. A five-to-one majority is hardly significant, for it could be obtained on over nine percent of occasions by random means, with no purposeful choice being made.

Thinking of the same kind applies to any situation in which individuals are being offered mutually exclusive choices. It even applies to general elections — ignoring the minor parties — but here the electorate is so large that even a small

(continued on next page)

Table 1.

How many choose coffee	Who has coffee	Number of ways
0	no one	1
1	A, B, C, D, E, F	6
2	A+B, A+C, A+D, A+E, A+F, B+C, B+D, B+E, B+F, C+D, C+E, C+F, D+E, D+F, E+F	15
3	A+B+C, A+B+D, A+B+E, A+B+F, 20 A+C+D, A+C+E, A+C+F, A+D+E, A+D+F, A+E+F, B+C+D, B+C+E, B+C+F, B+D+E, B+D+F, B+E+F, C+D+E, C+D+F, C+E+F, D+E+F	

```

10 HOME : PRINT TAB( 14) "THIS OR THAT?"
20 VTAB 5: INPUT "HOW MANY INDIVIDUALS IN TOTAL? ";N
30 IF N < 0 OR N > 30 THEN PRINT "NUMBER OUT OF RANGE, PLEASE RE-ENTER": GOTO 20
40 CALL - 868
50 VTAB 7: INPUT "HOW MANY IN ONE OF THE GROUPS? ";G
60 IF G < 0 OR G > N THEN PRINT "NUMBER OUT OF RANGE, PLEASE RE-ENTER": GOTO 50
70 PRINT
80 PRINT "CALCULATING"
90 IF G = N / 2 THEN P = 100: GOTO 220
100 IF G > N / 2 THEN S = N - G
110 C = 0
120 FOR J = 0 TO INT ((N - 1) / 2)
130 GOSUB 1000
140 IF J = G THEN CG = C
150 NEXT J
160 C = C * 2
170 CG = CG * 2
180 IF N / 2 = INT (N / 2) THEN NF = N / 2: GOSUB 2000
190 C = C + 1 / F / F
200 P = CG / C * 100
210 C = C * F
220 PRINT : PRINT "THE PROBABILITY OF OBTAINING A RESULT AS EXTREME OR MORE EXTREME THAN THIS IS:" : PRINT : PRINT TAB( 17) INT (P + .5); " %"
230 END
1000 NF = J: GOSUB 2000
1010 D1 = F
1020 NF = N - J: GOSUB 2000
1030 D2 = F
1040 C = C + 1 / D1 / D2
1050 RETURN
2000 F = 1: IF NF = 0 THEN RETURN
2010 FOR K = 1 TO NF
2020 F = K * F
2030 NEXT K
2040 RETURN

```

```

JRUN
      THIS OR THAT?
HOW MANY INDIVIDUALS IN TOTAL? 15
HOW MANY IN ONE OF THE GROUPS? 4

CALCULATING

THE PROBABILITY OF OBTAINING A RESULT AS EXTREME OR MORE EXTREME THAN THIS IS:

      12 %

```

Applesoft Basic program and sample run.

A question of significance

(continued from previous page)

majority is significant. It is when only a few individuals are making a choice that you need to assess more precisely the effects of random or partly random factors.

Experiments in animal behaviour are another instance in which it is essential to take possible randomness into account. It was in this setting that tests for significance were first designed. You often have only a few animals to use, so randomness plays a relatively large and disturbing part in the result. The same problem arises in other kinds of scientific and medical experiment — even those which do not involve individuals making a deliberate choice.

The alternative could be “does the patient recover or not?” or “does this drug kill the pathogen or not?” The criterion is that there must be two mutually exclusive outcomes with an apparently equal chance of either outcome.

Working out the odds for a large number of individuals is extremely tedious, which is where a micro is a great help. You have to find out in how many ways it is possible to partition the individuals into two groups. A group of n individuals partitioned into two groups can be represented by $x:y$, where $x + y =$

Table 2.

Partition	No. of ways
0:n	$n!/0!(n-0)! = 1$
1:n-1	$n!/1!(n-1)! = n$
2:n-2	$n!/2!(n-2)!$
and so on down to	
n-2:2	$n!/(n-2)!2!$
n-1:1	$n!/(n-1)!1! = n$
n:0	$n!/(n-0)!0! = 1$

n . For example two out of six directors taking coffee and four taking tea can be represented by 2:4.

The different partitions and the calculation of the numbers of ways are shown in table 2.

The symbol $!$ means factorial. For example, $5!$ means $5 \times 4 \times 3 \times 2 \times 1$. Unexpectedly, $0!$ is 1. The expressions in table 2 are all of the form

$$n!/g!(n-g)!$$

where g is the number of individuals making one choice and $(n-g)$ is the number making the other choice. The micro has to work out all these terms and add them to find out how many different partitions are possible.

Since the table is symmetrical about the halfway line, the micro only has to work out the top half and double the result. If n is even, there is a row halfway down the table for $n/2:n/2$. Picking out half the individuals in all possible ways to put into one group automatically picks out the other half to go into the other group, so this partitioning is added in only once.

While the computer is summing all these expressions, it also sums those expressions

which refer to partitions as extreme or more extreme than the one being tested. If one of the six directors chooses differently from the others you need to sum the expressions for 0:6 and 0:5 and then double the sum. You can then work out the probability of getting a majority decision of five or more out of six according to the formula.

$$\frac{((\text{number of ways for } 0:6, 0:5, 5:0, 6:0) + (\text{total number of ways})) \times 10 \text{ percent}}$$

The $n!$ in the denominators of each quantity cancel out, so there is no need to evaluate it.

Listings are provided for the Apple II and the BBC Microcomputer. The Apple II version requires n to be more than 2 and not more than 30. Calculating factorials greater than $33!$ causes an overflow error. Since the test is not of great interest when numbers are larger than 30 this is no disadvantage.

N is the number of individuals observed and G is the number in one of the groups. Line 90 disposes of one obvious result without calculation. Line 100 converts G to be the number of the smaller group. Lines 120 to 150 run through the possible partitions, except equipartition when N is even.

The program uses the subroutine beginning at line 1000 to work out the expression

$$1/G!(N-G)!$$

for each value of G in turn, and accumulates their total; the subroutine beginning at line 2000 calculates the factorials required. NF is the number for which the factorial is to be calculated and F is the factorial. In line 140 the subtotal of values up to and including G is registered as CG .

The totals C and CG are then doubled in lines 160 and 170. If N is even, a value for the partition $N/2:N/2$ is then added to the total obtained in lines 180 to 200. Line 210 calculates probability P as a percentage.

The percentage is rounded off to the nearest whole number; if you are interested in long odds you could alter the proposal to print out any number of decimal places. The sample run might have been used to assess the results of asking 15 breakfasters whether they would prefer kipper or haddock. The fact that 11 take kippers does not support the belief that breakfasters in general prefer kippers. With a purely random selection, there is a 12 percent chance that the number disagreeing with the majority will be four or fewer. A majority of 11 to four means very little.

Figure 2 shows that the ways in which 10 or more people can choose coffee — or kippers — at random is six percent of the total number of ways. The program gives an answer of 12 percent as in line 170 it doubles the numbers relating to the shaded area before working out the percentage. Which result you take depends on what you want to know.

If you want to know the probability of

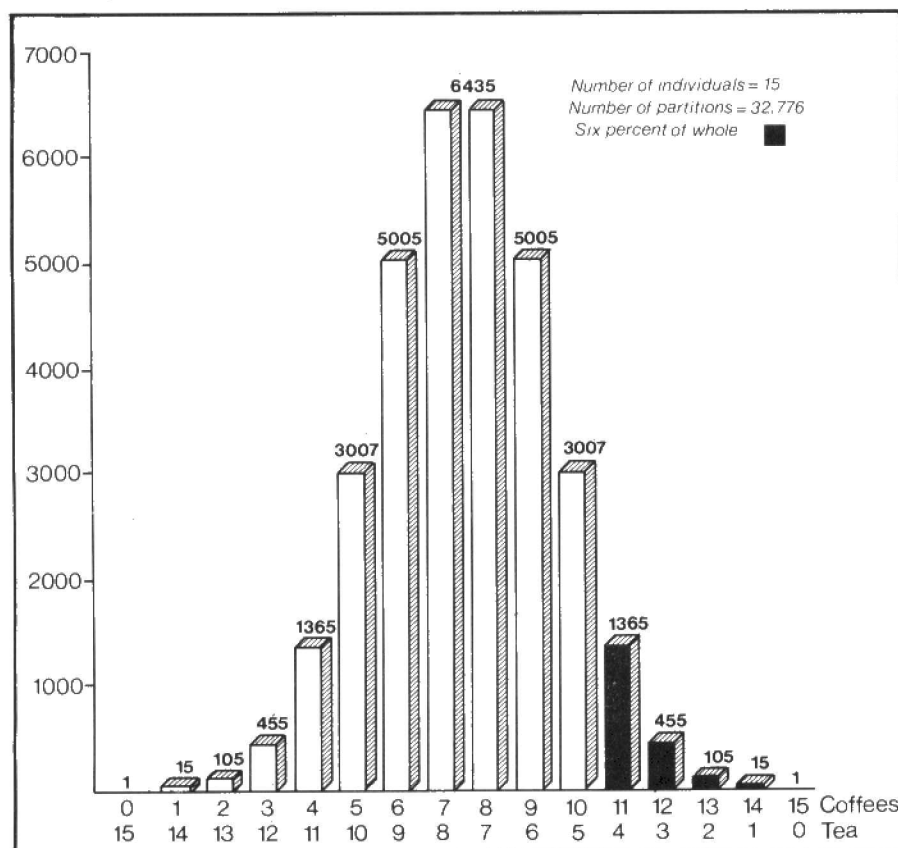


Figure 2. There is a six percent chance of only four people out of 15 taking tea.

getting any extreme result — either lots more coffees than teas or lots more teas than coffees — then take into account both tails of the distribution of figure 2. This is the usual procedure, as there is generally no reason for expecting a strong majority. After all, you are assuming that people like tea or coffee equally well. If a majority is found it is usually explained away as a random occurrence by saying, for example: "You could have got this majority by tossing heads or tails: it does not prove that people prefer coffee — or kippers".

There may be occasions on which you believe there is a preference and are trying to prove it. Then you say, for example: "11 out of 15 choose coffee; at random, 11 or more in favour of coffee occurs on only six occasions out of 100, so possibly this shows that there is a preference. The evidence is not strong, for there is still the six percent chance of it being a random result. But as figure 2 shows, the numbers of ways decrease sharply towards the tail of the distribution. If 12 choose coffee, the area to the right is only two percent of the whole. Just one more coffee drinker should make you feel much more secure in believing that coffee is preferred.

The BBC program is in principle the same as the Apple version but makes use of the special facilities available on the BBC machine. The calculation of each value of the fraction.

1/group!(total-group)!

```

10 CLS: PRINTTAB(13)"This? - or T
hat?"
20 INPUTTAB(0,5)"How many individ
uals in total",total
30 IF total<2 OR total>30 THEN PR
INT"Number out of range, please re-
enter":GOTO20
40 PRINTSPC(40)
50 INPUTTAB(0,7)"How many in one
of the groups",group
60 IF group<0 OR group>total THEN
PRINT"Number out of range, please r
e-enter":GOTO50
70 PRINTSPC(80)
80 PRINT"Calculating"
90 IF group = total/2 THEN Probab
ility = 100: GOTO 210
100 IF group > total/2 THEN group
= total - group
110 comb = 0
120 FOR partition = 0 TO INT((tota
l-1)/2)
130 comb = comb + FNfractions(part
ition)
140 IF partition = group THEN grou
pcomb = comb
150 NEXT partition
160 comb = comb*2
170 groupcomb = groupcomb*2
180 IF total/2 = INT(total/2) THEN
comb = comb + 1/FNfact(total/2)*2
190 probability = groupcomb/comb*1
00
200 PRINT:PRINT"The probability of
obtaining a result as extreme or mor
e extreme than this is:"PRINTTAB(9,
16) INT(probability + .5);" %"
210 END
1000 DEF FNfractions(partition)
1010 = 1/FNfact(partition)/FNfact(t
otal-partition)
2000 DEF FNfact(number)
2010 IF number = 1 OR number = 0 TH
EN =1 ELSE = number*FNfact(number-1)

RUN
This? - or That?
How many individuals in total?15

How many in one of the groups?4

CALCULATING
The probability of obtaining a resul
t as extreme or more extreme than thi
s is:
12 %

```

BBC Basic program and sample run.

is performed by a function FNfractions, defined at line 1010, which uses function FNfact at line 2010 to calculate the factorials.

The total number of ways, Comb, is accumulated at line 130. At line 140 this cumulative value is assigned to the variable, Groupcomb, when the partition being evaluated is the same as the partition observed. Cumulative totals are doubled at line 190 and the single addition is made

at line 180 in the case when the number of individuals is even.

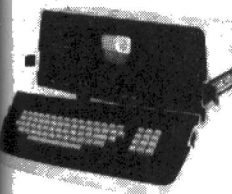
References:

Coffee, Tea or Me? by Trudy Baker and Rachel Jones. Corgi Books, 1967.
Longman Statistical Utility by Owen Bishop. Longman Microsoftware, 1983.
Statistics for Biology by Owen Bishop. Longman, (4th edition 1983).

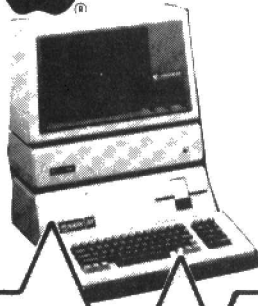
MICRO-RENT

from KEYBOARD HIRE

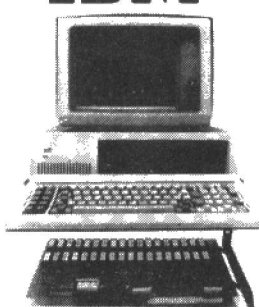
OSBORNE 1



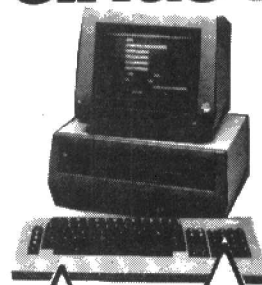
Apple



IBM



SIRIUS 1



MR

•607 8797•8

Totally flexible rentals of OSBORNE, APPLE, IBM and SIRIUS hardware, software and peripherals from under £3.50 per day (3-month rental, Osborne)

KEYBOARD HIRE LIMITED 176 BARNBURY ROAD LONDON N1 0ER

• Circle No. 195

Check-out network

At Orpington's Walsingham School they use a Research Machines micro to simulate a point-of-sale terminal. Hewan Ormson explains how it's done.

MANY LARGER SHOPS and supermarkets are installing electronic point-of-sale terminals, POSTs, which are generally on line to a central computer. There are several different methods of data capture, including: keyboard input of a code number; optical character reader, or light wand; magnetic stripe reader; laser scanner, or bar code reader; merchandise ticket reader, or kimball tag.

British Home Stores has branches in most major shopping areas, and they all have electronic POST. BBC Radio includes a programme in the "Computers in the Real World" series which examines this system. Each POST is connected to an in-store minicomputer, or an area minicomputer. The minicomputer holds all the data relating to the stock using disc storage. The minicomputer is joined by telephone line to a mainframe at BHS headquarters — see figure 1.

Everything sold in the store is given a six-

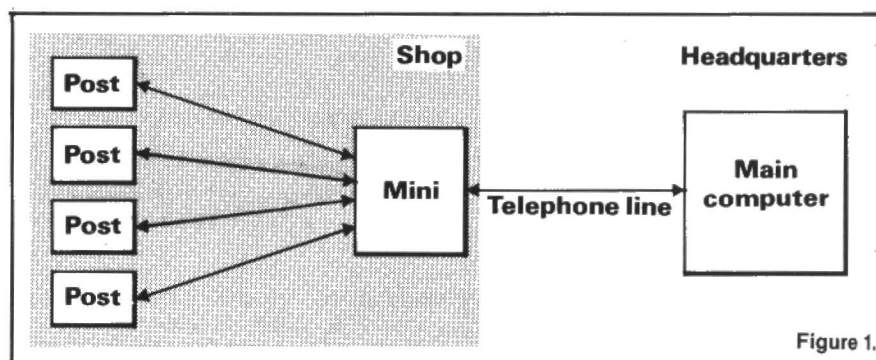


Figure 1.

digit code number. The prices are shown on the shelves rather than on each individual item. When a customer buys an article he or she takes it to a POST. The POST operator types in the code; the POST finds the item details from the minicomputer and displays the price. The operator keys in any other items and the POST totals the prices. The

operator keys in the amount of cash tendered, the POST displays the change due and prints out a receipt.

The receipt typically contains the following information: the type, price and number of each item purchased; the total cost; the amount of cash tendered and the change given; the date, etc. The

```

10 REM *****
20 REM *****
30 REM ***          ***
40 REM ***      SHOP      ***
50 REM ***  Version 1.1  ***
60 REM ***          ***
70 REM *** Demonstrate POST ***
80 REM ***          ***
90 REM ***  By H.Ormson  ***
100 REM ***   June 1983   ***
110 REM ***          ***
120 REM *****
130 REM *****
140 REM
150 CLEAR 1000: DIM C(20), I$(20), P$(20)
160 ON BREAK GOTO 1660
170 PUT 12,21
180 ?"                      SHOP
190 ??:?:?:?:?:?
200 A$=GET$(0)
210 INPUT "Type the date: ", D$
220 REM *Check that only RETURN not pr
    essed*
230 IF D$="" THEN PUT 11: GOTO 210
240 REM *Read in data*
250 RESTORE
260 FOR C=1 TO 20
270 READ C(C), I$(C), P$(C)
280 NEXT C
  
```

```

290 PUT 31
300 REM *Set up display*
310 ??:?:?:?:? The Walsingham S
    perstore"
320 ?"
    "
330 ??:?:? "Type the code number then p
    res RETURN"
340 ?"
    "
350 ??:? "      Code no:"
360 REM *Set up coords - initialise va
    rs*
370 X=33: Y=45: T=0: ZX=0
380 REM *Ask for code number*
390 FOR C=1 TO 4
400 PUT 22, Y, X, "Item", STR$(C), " ": INPUT
    T"", C1$
410 REM *Erase previous incorrect ent
    y - 43 blanks*
420 IF ZX=1 THEN PUT 22, Y, 49, "
    "
    X=0
430 REM *Find corresponding article*
440 FOR Z=1 TO 20
450 IF " "+C1$=STR$(C(Z)) THEN 510 ELSE
    E NEXT Z
460 REM *Give error message, set ZX=1
    10 blanks*
  
```

Figure 2.

The Computer in Shops

A. Copy the diagram showing how POSTS are connected to a computer.

B. Copy out these sentences and fill in the gaps.

- In a lot of big shops, tills are being replaced with _____.
- These are connected to a _____.
- Every item in the shop is given a code. Three things the code might tell us are
 - _____
 - _____
 - _____
- The price is not shown on each item. Instead it is shown on the _____.
- This may cause problems because _____.
- A receipt is printed out. This will show
 - _____
 - _____
 - _____
 - _____
- Two advantages of using computerised tills are
 - _____
 - _____
- Every night, the main computer automatically "speaks" to the _____ computer. This is done through a _____ line.
- The mini computer tells the main computer _____.

minicomputer keeps a record of all items sold. This information is automatically sent to the mainframe each evening. Further details of the system, follow-up notes and a filmstrip are available from BBC Publications.

The program listed complements the
(continued on next page)

The Walsingham Superstore

Receipt 7 JULY

691307	lamp	1	4.99
683303	sports bag	1	5.99
421096	light bulb	1	0.25
653692	shirt	1	6.50
Total			£ 17.73
Cash			£ 20.00
Change			£ 2.27

Thank you for your custom
Please call again

Figure 4.

SHOPPING LIST

Choose four items from the table below.
Make a shopping list.
Write down each item and its code number.
Type the code numbers for your items into the computer.
The computer will tell you the price of each item.
Write this on your shopping list.
The computer will work out the total cost.
Type in how much money you would give the assistant if you were the customer.
e.g. 5.00, 1.50, 0.95, 2.05.
Write this down.
The computer will print out a "receipt".
Fasten it into your book.

Item	Code	Item	Code
plug	943303	Clock	705869
toothpaste	185233	annual	536274
shampoo	246717	football book	837350
blouse	243304	doll	697261
sports bag	683303	shirt	653692
light bulb	421096	scarf	366890
shoes	174603	handkerchief	605466
gloves	276508	nightdress	504696
socks	191949	toy car	804379
lamp	691307	pyjamas	383036

Figure 3.

```
PUT 22,Y+1,40,"Code number wrong.
again"
PUT 22,Y,39,"      "
TX=1:60TO 400
REM *Print code & article*
PUT 22,Y,49,I$(Z):PUT 22,Y,63,"1":
22,Y,65,P$(Z)
REM *38 blanks*
PUT 22,Y+1,40,"      "
REM *increase total & Y coord*
Y=Y+1:T=T+VAL(P$(Z)):T$(C)=I$(Z):T
C=P$(Z):T(C)=C(Z)
NEXT C
PUT 22,50,57,"Total"
T=INT(T*100)/100
L=T
REM *Add trailing zeros etc*
GOSUB 1350
REM *Print total in correct positi
IF LEN(A1$)=6 THEN PUT 22,50,64,A1
TO 650
PUT 22,50,65,A1$
REM *Ask for cash*
PUT 22,52,57,"Cash?      ":INPUT "", C
IF CASH$="" THEN 650
REM *Check input for alpha charact
```

```
ers*
680 FOR B1=1 TO LEN(CASH$)
690 IF ASC(MID$(CASH$,B1,1))>57 OR ASC
(MID$(CASH$,B1,1))<46 THEN PUT 22,52,36
,"Use numbers only
":GOTO 650
700 NEXT B1
710 CASH=VAL(CASH$):CASH=INT((CASH+.00
1)*100)/100
720 REM *Output cash in correct positi
on
730 L=CASH:GOSUB1350
740 IF LEN(A1$)=5 THEN A1$=" "+A1$
745 REM *erase previous entry - 40 bla
nks*
750 PUT 22,52,36,"      "
760 PUT 22,52,57,"Cash      ":?;A1$
770 REM *Too much cash offered?*
780 IF CASH <=99.99 THEN 830
785 REM *erase previous entry - 30 bla
nks*
790 PUT 22,52,57,"      "
800 PUT 22,52,36,"Too much. Try again
"
810 GOTO 650
820 REM *Calculate change*
```

(listing continued on next page)

(continued from previous page)

Radiovision program, but it can be used on its own to supplement work on computers in shops — see figure 2. It is written for a Research Machines 380-Z or 480-Z and simulates a POST used in British Home Stores. It occupies about 4K.

Pupils choose four items from the list in figure 3 and key in the code numbers. The screen displays the code, the item description and its price. It totals the prices and asks for the amount of cash tendered; the maximum allowed is £99.99. The amount of change due is displayed and an itemised receipt is printed — see figure 4. The program checks for illegal code numbers, too much or too little cash, alphabetic instead of numeric input, and

adds trailing zeros. In order to do all this code numbers and cash are entered as strings.

Twenty items are included in the data. This can be increased or decreased as

required. The code numbers are random numbers as generated by a 380-Z. They could be improved by making the first two or three digits represent a particular department and the last a check digit. □

Variable Description

CL)	item code number, six digit	C1\$	input code number
IS()	item name	T	total of goods purchased
PS()	item price	TS()	item stored here
AS	wait/trap key presses	T1\$()	price ready for
DS	date	T()	code printing
B1)		L	temporary store used in
C }	loop counters	A1\$ }	trailing zero routine
Z }		CASH\$	amount of cash tendered
X,Y	co-ordinates of item on screen	CASH	value of CASH\$
ZX	count: if 1 erase incorrect entry	CH	change to be given

(listing continued from previous page)

```

830 CH=CASH-T
840 REM *Remove any rounding errors*
850 CH=INT((CH+.001)*100)/100
860 IF CH<.01 AND CH>=.009 THEN CH=.01
:GOTO 910
880 REM *Check for too little cash*
890 IF CASH<T THEN PUT 22,52,36,"Not e
nough cash. Try again
" ELSE 910:REM *22 blanks*
900 PUT 22,52,65:INPUT "",CASH$:GOTO 68
0
910 PUT 22,53,65,"-----"
920 IF CH<1 THEN CH$=STR$(CH):CH$=LEFT
$(CH$,4):CH=VAL(CH$)
930 REM *Add trailing zeros etc*
940 REM *Output change in correct posi
tion*
950 L=CH:GOSUB 1350
960 IF LEN(A1$)=5 THEN A1$=" "+A1$
970 PUT 22,54,57,"Change ",A1$
980 PUT 22,55,65,"-----"
990 REM *Print receipt*
1000 LPRINT"The Walsingham Superstore"
1010 LPRINT:LPRINT"Receipt ";D$
1020 LPRINT
1030 FOR C=1 TO 4
1040 LPRINT T(C);TAB(10);T$(C);TAB(27)
;"1";
1050 IF LEN(T1$)=6 THEN LPRINT TAB(30)
;T1$(C) ELSE LPRINT TAB(31);T1$(C)
1060 NEXT C
1070 LPRINT
1080 L=T
1090 GOSUB 1350
1100 LPRINT TAB(20);"Total";TAB(27);"£
";
1110 IF LEN(A1$)=6 THEN LPRINT TAB(30)
;A1$ ELSE LPRINT TAB(31);A1$
1120 LPRINT
1130 L=CASH
1140 GOSUB 1350
1150 LPRINT TAB(21);"Cash";TAB(27);"£
";
1160 IF LEN(A1$)=6 THEN LPRINT TAB(30)
;A1$ ELSE LPRINT TAB(31);A1$
1170 LPRINT
1180 L=CH
1190 GOSUB 1350
1200 LPRINT TAB(19);"Change";TAB(27);"
£";
1210 IF LEN(A1$)=6 THEN LPRINT TAB(30)
;A1$ ELSE LPRINT TAB(31);A1$

```

```

1220 LPRINT
1230 LPRINT" Thank you for your custom
"
1240 LPRINT
1250 LPRINT" Please call again"
1260 LPRINT:LPRINT:LPRINT
1270 PUT 12
1280 ?"Take your receipt"
1290 ?:"?"-----
:"?:?
1300 A$=GET$(200):?:"?
1310 ?"Is there another customer? (Y/N
) ":A$=GET$(0):A$=GET$(0)
1320 IF A$="Y" OR A$="y" THEN 250
1330 IF A$="N" OR A$="n" THEN 1660 ELS
E PUT 11:GOTO1310
1340 REM *Subroutine for adding traili
ng zeros
1350 A1$=STR$(L):L1=LEN(A1$):J=INT(L)
1360 IF L1>=6 THEN 1430
1370 IF L1=5 AND MID$(A1$,3,1)<>". " TH
EN A1$=A1$+"0":GOTO 1430
1380 IF L1=4 AND J=0 THEN A1$=" 0"+RIG
HT$(A1$,3) :GOTO 1430
1390 IF L1=4 THEN A1$=A1$+"0":GOTO1430
1400 IF L1=3 AND MID$(A1$,2,1)="." THE
430
1420 IF L1=2 THEN A1$=A1$+"00"
1430 RETURN
1440 REM *Data - code no, item, price*
1450 DATA 943303,plug," 0.45"
1460 DATA 185233,toothpaste," 0.39"
1470 DATA 246717,shampoo," 0.40"
1480 DATA 243304,blouse," 4.75"
1490 DATA 683303,sports bag," 5.99"
1500 DATA 421096,light bulb," 0.25"
1510 DATA 174603,shoes,12.99
1520 DATA 276508,gloves," 3.00"
1530 DATA 191949,socks," 0.57"
1540 DATA 691307,lamp," 4.99"
1550 DATA 705869,clock,12.99
1560 DATA 636274,annual," 1.50"
1570 DATA 837350,football book," 2.25"
1580 DATA 697261,doll," 3.75"
1590 DATA 653692,shirt," 6.50"
1600 DATA 366890,scarf," 4.20"
1610 DATA 605466,handkerchief," 0.30"
1620 DATA 504696,nighdress," 8.50"
1630 DATA 804379,toy car," 1.75"
1640 DATA 382036,pyjamas," 6.30"
1650 REM *Finished*
1660 PUT 12,23
1670 END

```

- occasionally the district code contains an alpha as well as a numeric character before the space.

The only operator instruction that has to be given is to insert a space to denote the end of the first section of the postcode if it contains only one numeric character; the space is inserted automatically if it contains two numerics. The postcode, area, district, sector and street are stored as strings C\$(1), C\$(2), C\$(3), C\$(4) so that further validation can take place and to simplify their use as sorting keys.

Geographical location is the key to sorting your sales information, says John Locke.

Random-access disc files are also set up for postcode cross-reference, dealer information rep information, satisfied user information and product

```
FOR X = 1 TO 120: IF C$(1) <> CY$(X)
  THEN NEXT X
```



The loop is exited with the match number X, which is then used to access directly the postcode record. This in turn is used to access the rep information and all relevant dealer information. Up to three satisfied users are always used for an area, so the Satisfied Users file can be accessed for record numbers from $3X - 2$ to $3X$.

In this simple example, a printout can be made of enquiry number, postcode, all applicable dealers, names and addresses together with further selection information such as their sales and credit ratings, exclusion from certain products, etc., and up to three satisfied users' names and addresses. This list can be pinned to the original enquiry letter for the sales staff to make their selection of one dealer and one user. Alternatively, further automatic selection can take place. In either case, to output a sales letter only the enquirer's name and address — if not already stored — the product, dealer, rep and user numbers need to be inserted.

While the dealer file is being constructed, the postcodes covered by each dealer have to be inserted in the postcode cross-reference file so that these, through the array match, will write the dealer number in the first vacant field of that postcode record. Amendment and deletion of dealer records must also access and modify these fields. The whole process is complex but routine. It slightly slows entry, amendment or deletion of dealer data, but is not a significant overhead as sales enquiries are answered more often than records are updated.

A great deal of detail is available from the Post Office on postcodes, ranging from the complete address file on magnetic tape for £15,000 down to publications on post towns, valid sectors, etc. Magnetic-tape file extracts are available from £4.75 per thousand records, subject to minimum charges. Post zone files of codes and Ordnance Survey grid references are available from £6,000 on tape. Postcode maps are available from J Bartholomew & Sons Ltd, Geographia Ltd, and Postal Headquarters.

There are also advantages in sorting outgoing mail when bulk posting is used. The Post Office gives a rebate on bulk posting of second-class mail that has been pre-sorted according to post code. For example, 5,000 to 23,529 letters are given a 15 percent rebate on the postage paid if they are pre-sorted. These levels are shown in the Inland Compendium held by main post offices.

The level of rebate is dependent not only on the total number of letters mailed but also on the amount of pre-sorting carried out. A booklet will be available shortly from regional offices and head postmasters giving more details. The Post Office's Post Code Marketing Section emphasise the importance of involving the local Post Office in proposals for bulk posting at an early stage.

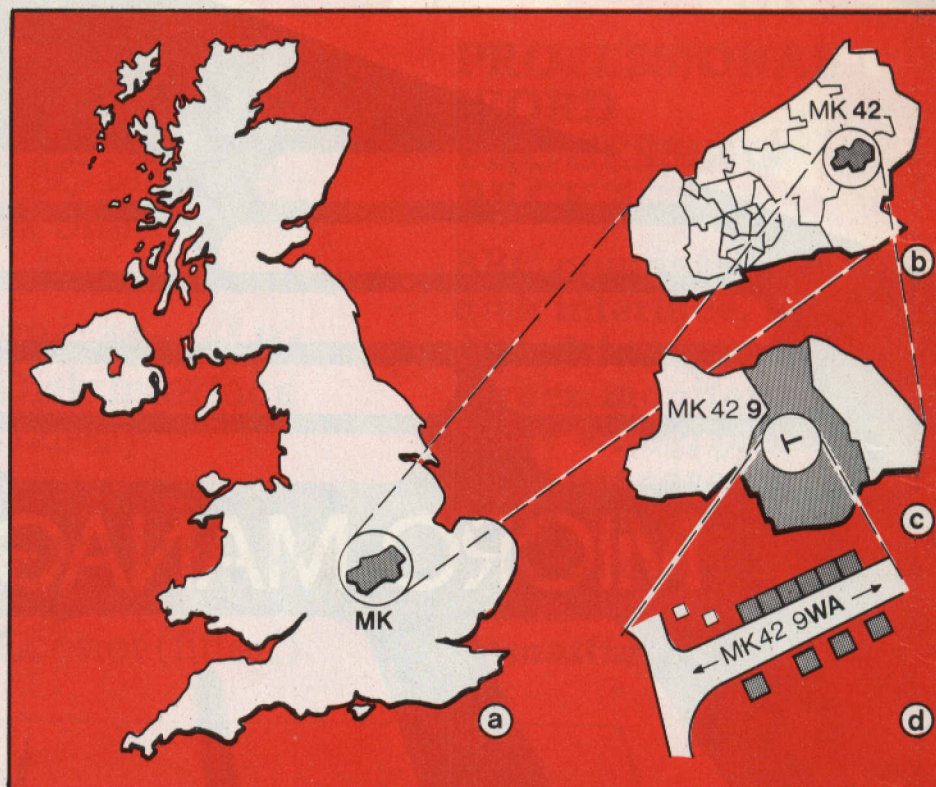


Figure 2. Most business letters include the postcode in their address which can form the basis of a geographical sort.

Postcode listing.

```
200 CLS:PRINT"ENTER POSTCODE":Z=1:
PRINTCHR$(14):FORX=1TO4:C$(X)="" :NEXTX
210 Z$=INKEY$:IFZ$="" THEN210
220 IFASC(Z$)>64ANDASC(Z$)<91THENGOTO260
230 IFASC(Z$)>48ANDASC(Z$)<58THENGOTO320
240 IFZ$=CHR$(8)THENGOTO 390
250 IFZ$=CHR$(32)ANDZ=2THENZ=3:PRINTZ$::
GOTO210
260 IFZ=2ORZ=3THENGOTO210
270 IFLEN(C$(Z))<2THENGOTO360
280 Z=Z+1:IFZ<5THENGOTO210
290 PRINT:PRINT"POSTCODE= ";C$(1)+C$(2)
+CHR$(32)+C$(3)+C$(4)
300 INPUT"IS THIS CORRECT";K$:IFK$="Y"
THENK=1ELSEIFK$="N"THENK=2ELSEPRINT"ENTER
'Y' OR 'N' ONLY":GOTO300
310 ONK GOTO410,200
320 IFZ=1ANDC$(1)<>" "THENZ=2
330 IFZ<2ANDZ<3THENGOTO210
340 IFZ=3THENGOTO370
350 IFLEN(C$(2))<1THENPRINTCHR$(32)::Z
=3:GOTO380
360 PRINTZ$::C$(Z)=C$(Z)+Z$:IFLEN(C$(4))
=2THENGOTO290ELSEGOTO210
370 IFLEN(C$(3))<0THENGOTO280
380 PRINTZ$::C$(3)=C$(3)+Z$:GOTO280
390 PRINTCHR$(8)::X=LEN(C$(Z)):IFX-1=0
THENC$(Z)="" :Z=Z-1:GOTO210
400 C$(Z)=LEFT$(C$(Z),X-1):GOTO210
410 END
```

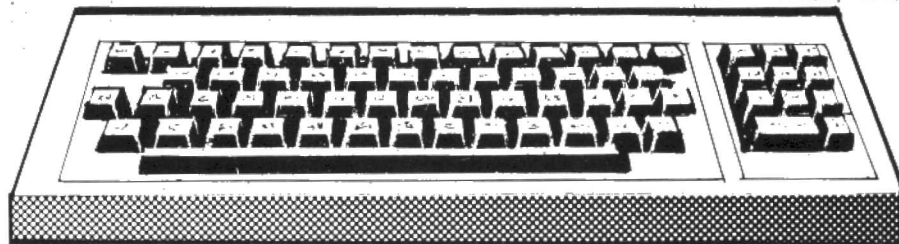

Open File

This regular section of *Practical Computing* appears in the magazine each month, incorporating Tandy Forum, Apple Pie, Sinclair Line-up and other software interchange pages.

Open File is the part of the magazine written by you, the readers. All aspects of microcomputing are covered, from games to serious business and technical software, and we welcome contributions on CP/M, BBC Basic, Microsoft Basic, Apple Pascal and so on, as well as the established categories.

Contributors receive £30 per published page and pro rata for part pages, with a minimum of £6. Send contributions to: Open File, *Practical Computing*, Quadrant House, The Quadrant, Sutton, Surrey SM2 5AS.

BBC Bytes: Sequencer routine; VDU23 definition; Fill routine for graphics; Face — a computerised joke; Box spin	141
Apple Pie: Disc patch program; Sub-exterminator game; HGR strings for graphics display	149
Research Machines Review: Helicopter rescue game	154
Newbrain: General-purpose graph; Format. 40 for column listings; Multiple screen display routines	156
Sinclair Line-up: User-defined I/O routines; Smooth scroll routine; Side scroll routine; Bridge hand tuition; Correlation coefficient to help anyone using statistics	160
End of File: Basic listing program for Sharp MZ-80B; High-resolution dump for Dragon 32; Nascom as terminal	164



Guidelines for contributors

Programs should be accompanied by documentation which explains to other readers what your program does and, if possible, how it does it. It helps if documentation is typed or printed with double-line spacing — cramped or handwritten material is liable to delay and error.

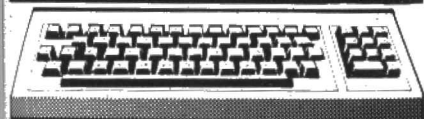
Program listings should, if at all possible, be printed out. Use a new ribbon in your

printer, please, so that we can print directly from a photograph of the listing and avoid typesetting errors. If all you can provide is a typed or handwritten listing, please make it clear and unambiguous; graphics characters, in particular, should be explained.

PLEASE send a cassette or disc version of your program if at all possible. It will be returned after use. For CP/M programs use IBM-format 8in. floppy discs.

BBC BYTES

by John Harris



Sequencer

Adrian Roe of Ilkley, whose Lightcycle game was published on page 164 of the

May issue, has submitted a program which I cannot claim to understand for all my efforts to do so. The program, I am told, transforms my machine into a sequencer with key, octave, tempo and bass control over a basic note pattern input as data at line 220.

Since the program uses the negative Inkey feature, interesting effects can be produced by holding down more than one key. Try holding the speed-up and pitch keys together, for example.

Sequencer.

```

10 REM***SEQUENCER BY A.ROE JAN'83
***
20 REM*****VERSION U.2*****
***
30 MODE2
40 VDU23:8202:0;0;0;
50 PROCINIT
60 PROCKEYS
70 RESTORE
80 READPIT:IFPIT=1000 GOTO70
90 IFINKEY(-122) BASS=BASS+48:SOU

```

```

ND2,2,BASS,1:SOUND3,2,BASS,1
100 IFINKEY(-26) BASS=BASS-48:SOU
ND2,2,BASS,1:SOUND3,2,BASS,1
110 IFINKEY(-54) KEY=KEY+4:BASS=BA
SS+4:SOUND2,2,BASS,1:SOUND3,2,BASS,1
120 IFINKEY(-51) KEY=KEY-4:BASS=BA
SS-4:SOUND2,2,BASS,1:SOUND3,2,BASS,1
130 IFINKEY(-103) TEMPO=TEMPO+1
140 IFINKEY(-104) TEMPO=TEMPO-1
150 IFINKEY(-58) OCT=OCT+48
160 IFINKEY(-42) OCT=OCT-48
170 SOUND0,-10,4,1:SOUND1,1,PIT+OC
T+KEY,1
180 TIME=U:REPEAT UNTIL TIME>=TEMP
O
190 IFINKEY(-82)ANDPIT=9 THENTIME=
U:REPEAT UNTIL TIME>=200:GOTO240
200 GOTO80
210 REM***NOTE SEQUENCE**
220 DATA9,37,49,57,49,37,29,37,100
U
230 REM***FINISH WITH 1000**
240 MODE7
250 *FX15,0
260 SOUND2,0,0,0:SOUND3,0,0,0
270 END
280 DEFPROCINIT
290 VDU23,230,8,4,2,255,2,4,8,0
300 VDU23,231,16,32,64,255,64,32,1
(listing continued on next page)

```


VDU23 definition

Following on from Mr P Davidson's eight-by-eight pixel manipulation program — published on page 136 of the February issue — this implementation by John Wilson of Benfleet, Essex provides far wider facilities. Up to 40 characters may be defined at a time on a cross-linked 10-by-four set of eight-by-eight pixels.

The resulting character definitions may be inverted from black to white and vice versa, Spooled for *Exec retrieval into a Basic source program, edited, and viewed in any mode. It is difficult to think of anything missing from the functions.

Operating instructions are written into the program, with a function-select screen

and subsequent prompts. The pixel map is represented as a grid of dots and an X cursor which is moved by the cursor-control keys. Pixels are selected with Copy and deleted with Delete. An agreeable feature is that the cursor controls can be held concurrently with select and deselect, providing rapid spreading or removal of lines in any of the eight cardinal directions.

If you have OS 0.1 you will not be able to employ the full 10 by four generated character set in your own Basic programs as no provision is made within the OS to "explode" the memory allocation for user-defined characters. OS 1.0 and above does cater for this, as shown on pages 427 and 428 of the *User Guide* under *FX20.

VDU23 definition.

```

10 REM
20 REM Character Generator - V-I
I
30 REM
40 REM for the BBC Micro Model B
50 REM
60 REM by J.R.Wilson April 1983
70 REM
80 DIM M% 256U, T% 32U, Z% 7
90 X% = 0: Y% = 0
100 REM Make cursor keys give cod
es
110 *FX4,1
120 REM Trap any errors
130 ON ERROR GOTO 670
140 D=0
150 W=0
160 XM=0: YM=0
170 REM
180 REM Display menu
190 REM
200 MODE4
210 VDU4
220 HIMEM=3000
230 VDU 23,224,255,255,255,255,255
240 VDU23,225,0,0,0,24,24,0,0
250 COLOUR129: COLOURU
260 PRINT STRING$(240," "); TAB(U,
U);
270 PRINT " CHARACTER
GENERATOR "
280 PRINTTAB(15,4); ">MENU<";
290 COLOUR1: COLOUR128
300 PRINT " 1 Enter character
r"
310 PRINT " 2 Edit character
"
320 PRINT " 3 Spool character
r to cassette"
330 PRINT " 4 View character
"
340 PRINT " 5 Invert character
er"
350 PRINT " 6 List codes"
360 PRINT " 7 Operating syst
em call"
370 PRINT " 8 End program"
380 PRINT "Press the number the
n RETURN ";
390 A=FNCH(8)
400 CLS
410 IF A=1 THEN MODEU: PROCENTER
420 IF A=2 THEN MODEU: HIMEM=3000
: PROCEDIT
430 IF A=3 THEN PROCCHANGE: PROCSP
UOL
440 IF A=4 THEN PROCCHANGE: PROCHO
DE: MODE HIMEM=3000: PROCVIEW
450 IF A=5 THEN PROCINVERT
460 IF A=6 THEN PROCEND
470 IF A=6 THEN PROCCHANGE: PROCLI
ST
480 IF A=7 THEN PROCoscall
490 REM Clear keyboard buffer
500 *FX15,1
510 GOTO 200
520 REM
530 REM Enter a number routine
540 REM
550 DEF FNCH(C)
560 LOCAL E,A
570 E=-1
580 A=GET: IF A=15 AND E<>-1 THEN
=E
590 IF C=10 AND A=65 THEN VDU A,6
:E=10: GOTO 580
600 IF A<49 OR A>48+C THEN 580
610 PRINT CHR$(A); CHR$(8);
620 E=A-48
630 GOTO 580
640 REM
650 REM Trap any errors
660 REM
670 IF ERR=17 THEN 200
680 REPORT
690 IF ERR>200 THEN PROCSPACE(31)
: GOTO 200
700 PRINT " in line "; ERL
710 GOTO 770
720 REM
730 REM End the program
740 REM
750 DEF PROCEND
760 VDU22,7: REM Change to mode 7
770 PRINTTAB(U,5)
780 REM Give the cursor keys thei
r usual function
790 *FX4,0
800 END
810 REM
820 REM Enter a character into me
mory
830 REM
840 DEF PROCENTER
850 PRINT "Number of characters
wide (1 to 9 or A ; where A=10 wid
e) ";
860 W=FNCH(10)
870 PRINT "Number of characters
deep (1 to 4) ";
880 D=FNCH(4)
890 CLS
900 XM=W*8-1
910 YM=D*8-1
920 FOR IX=0 TO YM*8U+8U
930 NX?IX=0
940 NEXT
950 X% = 0: Y% = 0
960 PROCEDIT
970 ENDPROC
980 REM
990 REM Change the picture into b
inary
1000 REM
1010 DEF PROCCHANGE
1020 PRINT "Please wait a few sec
onds."
1030 FOR IX=0 TO D-1
1040 FOR JX=0 TO W-1
1050 FOR KX=0 TO 7
1060 ?(TX+KX+JX*8+IX*8U)=0
1070 FOR LX=0 TO 7
1080 IF ?(MX+JX*8+LX+((IX*8+KX)*8U
))=1 THEN ?(TX+KX+JX*8+IX*8U)=?(TX+K
X+JX*8+IX*8U)+2*(7-LX)
1090 NEXT LX,KX,JX,IX
1100 ENDPROC
1110 REM
1120 REM Edit the picture
1130 REM
1140 DEF PROCEDIT
1150 IF YN=0 OR XN=0 THEN ENDPROC
1160 IF YN=31 AND XN=79 THEN VDU5:
REM Stop the screen from scrolling u

```

(listing continued from previous page)

```

6,0
310 VDU23,232,16,56,84,146,16,16,1
6,0
320 VDU23,233,16,16,16,146,84,56,1
6,0
330 ENVELOPE2,1,0,0,0,0,0,127,0,
0,0,75,75
340 ENVELOPE1,1,0,0,0,0,0,127,-1
0,-10,-1,126,60
350 OCT=0: BASS=9: TEMPO=16: KEY=0
360 UD$=" = DOWN - UP = "
370 ENDPROC
380 DEFPROCKEYS
390 COLOUR1
400 PRINTTAB(3,1)"TEMPO CONTROL"TA
B(1,4)"<";UD$;">": PROCLINE(5)
410 COLOUR2: PRINTTAB(2,7)"SEQUENCE
CONTROL"TAB(1,10)CHR$23UD$CHR$232:
PROCLINE(11)
420 COLOUR3: PRINTTAB(4,13)"BASS CO
NTROL"TAB(1,16)CHR$23UD$CHR$230: PRO
CLINE(17)
430 COLOUR5: PRINTTAB(4,19)"KEY CON
TROL"TAB(1,22)"D";UD$;"U": PROCLINE(2
3)
440 COLOUR6: PRINTTAB(4,25)"STOP CO
NTROL"TAB(3,28)"HOLD DOWN 'S' "
450 ENDPROC
460 DEFPROCCLINE(L)
470 COLOUR7: FOR I=0 TO 19: PRINTTAB(1,
L)" "I: NEXT
480 ENDPROC

```

```

P
1170 FOR IX=0 TO YM
1180 FOR JX=0 TO XM
1190 PRINT TAB(JX,IX);
1200 IF ?(HX+JX+IX*8U)=0 THEN VDU
225 ELSE VDU 224
1210 NEXT JX,IX
1220 VDU 23,8202;U;0;0;: REM Make c
ursor invisible
1230 IF X% = 79 AND Y% = 31 THEN VDU5,
9,127 ELSE VDU4
1240 PRINT TAB(X%,Y%);"X";
1250 AS=GET$
1260 PRINTTAB(X%,Y%);
1270 IF X%<>79 OR Y%<>31 THEN 1310
1280 VDU9,127,30
1290 IF ?MX=1 THEN VDU 224 ELSE VD
U 225
1300 PRINTTAB(X%,Y%);
1310 IF ?(MX+X%+Y%*8U)=1 THEN VDU
224 ELSE VDU 225
1320 REM Clear the keyboard buffer
1330 *FX15,1
1340 REM Test the cursor keys
1350 IF INKEY(-26) THEN X%=X%-1
1360 IF INKEY(-122) THEN X%=X%+1
1370 IF INKEY(-42) THEN Y%=Y%+1
1380 IF INKEY(-58) THEN Y%=Y%-1
1390 REM Keep the cursor on the pi
cture
1400 IF Y%=-1 THEN Y%=YM
1410 IF Y%>YM THEN Y%=0
1420 IF X%=-1 THEN X%=XM
1430 IF X%>XM THEN X%=0
1440 REM Delete a block
1450 IF INKEY(-90) THEN ?(NX+X%+Y%
*80)=0
1460 REM Leave a block
1470 IF INKEY(-100) THEN ?(MX+X%+Y
%*80)=1
1480 GOTO 1250
1490 REM
1500 REM What mode do you want to
display the character in ?
1510 REM
1520 DEFPROCMODE
1530 PRINT TAB(0,10);"What mode fo
r display ? ";
1540 INPUT "M
1550 IF M<0 OR M>6 OR M<>INT THEN
CLS: GOTO 1530
1560 ENDPROC
1570 REM
1580 REM Display the character
1590 REM
1600 DEF PROCVIEW
1610 FOR IX=0 TO D-1
1620 FOR JX=0 TO W-1

```

A Fill routine.

5 REM The variables in the testbed are: cursor at x,y; v=4 for cursor moving, 5 for cursor drawing; use cursor control keys and joystick if available else REM out line 130 and remove the OR((ADVAL(U)AND5)<>0) from line 70

10 DATA RED, GREEN, YELLOW, BLUE, MAGENTA, CYAN, WHITE

20 MODE7:PRINT""The colour codes used are:"";FOR I=1 TO 7:READ col\$:PRINT CHR\$(128+I);"";I;" for ";col\$;SPC(10-LEN(col\$));CHR\$(157);CHR\$(128+I):NEXT I:PRINT""and U for black""

30 MODE3:PRINT""Controls are:"";B or F change the background or foreground to the number keyed after""C clears to MODE number keyed after""SPACE or Trigger toggles between MOVE and DRAW""Arrows and joystick work""COPY fills"

40 PRINT""K sets a border""G goes to it""O sets an orientation""N a number of sides""M draws a polygon""T writes text 'til RETURN""A varies the GCOL option U plot 1 OR 2 AND 3 EOR 4 Invert""press space to go on..."

50 PROC Gk(""):x=500:y=500:edgex=x:edgex=y:rad=U:v=4:gtype=U:orient=U:sides=36:forex=7:backx=0:MODE 2:MOVE x,y:VDU5:PROCcoeff:REPEAT:FX15,0

60 IF INKEY(-68) THEN forex=VAL(GETS):GCOLgtype,forex ELSE IF INKEY(-101) THEN backx=VAL(GETS):GCOLgtype,backx+12

8 ELSE IF INKEY(-83) THEN PROC Gk("012"):MODE VAL(a\$):MOVE x,y:VDU5:PROCcoeff:GCOLgtype,forex:GCOLgtype,128+backx

70 tog=INKEY(-99) OR ((ADVAL(U)AND5)<>0):IF tog AND v=4 THEN v=5:PROC delay(10) ELSE IF tog AND v=5 THEN v=4:PROC delay(10) ELSE IF INKEY(-36) THEN PROC delay(25)

90 IF INKEY(-58) THEN PROC Gk("01234"):gtype=VAL(a\$):GCOLgtype,forex:GCOLgtype,128+backx

100 IF INKEY(-42) THEN PROC Gk("01234"):gtype=VAL(a\$):GCOLgtype,forex:GCOLgtype,128+backx

110 IF INKEY(-26) THEN PROC Gk("01234"):gtype=VAL(a\$):GCOLgtype,forex:GCOLgtype,128+backx

120 IF INKEY(-122) THEN PROC Gk("01234"):gtype=VAL(a\$):GCOLgtype,forex:GCOLgtype,128+backx

130 PROC Gk(FNad(ADVAL(2)),FNad(ADVAL(1))):REM this line out if you don't want joystick control

140 PROC Gk:PROC Gk:IF INKEY(-106) THEN PROC Gk(x,y,backx):MOVE x,y:v=4

150 UNTIL FALSE

160 DEF FNad(A\$):IF A\$<25000 THEN INT((A\$-25000)/1000) ELSE IF A\$>40000 THEN INT((A\$-40000)/1000) ELSE 0

170 DEF PROC C:PL0T2,12,12:PL0T2,-24,0:PL0T2,0,-24:PL0T2,24,0:PL0T2,0,24:PL0T2,-12,-12:ENDPROC

180 DEF PROC Cpolygon:IF rad=U THEN rad=FNhype

190 startx=x+COS(orient)*rad:start y=y+SIN(orient)*rad:MOVE startx,start y:FOR angle=orient TO orient+2*PI STEP 2*PI/sides:newx=x+COS(angle)*rad: newy=y+SIN(angle)*rad:DRAW newx,newy:NEXT:IF newx<>startx OR newy<>starty THEN DRAW startx,starty

200 MOVE x,y:ENDPROC

210 DEF PROC C(XI,YI):IF x+XI<1280 AND x+XI>0 THEN x=x+XI

220 IF y+YI<1024 AND y+YI>0 THEN y=y+YI

230 PL0T v,x,y:ENDPROC

240 DEF PROC Ccoeff:PROC Cosbyte(&97,0,1):PROC Cosbyte(&97,1,32):ENDPROC

250 DEF PROC Ccon:PROC Cosbyte(&97,0,1):PROC Cosbyte(&97,1,96):ENDPROC

1630 FOR LX=0 TO 7

1640 ZX=LX*(TX+LX+JX*8+IX*80)

1650 NEXT LX

1660 VDU23,224,ZX,ZX*1,ZX*2,ZX*3,ZX*4,ZX*5,ZX*6,ZX*7

1670 PRINT TAB(JX*5,IX*5);CHR\$(224);

1680 NEXT JX,IX

1690 IF M=3 OR M=6 THEN PROCSPACE(24):ENDPROC

1700 IF M=2 OR M=5 THEN PROCSPACE(30):ENDPROC

1710 PROCSPACE(31)

1720 ENDPROC

1730 REM

1740 REM Press the SPACE BAR to continue

1750 REM

1760 DEF PROCSPACE(YX)

1770 PRINT TAB(0,YX);"Press the SPACE BAR to continue."

1780 REPEAT UNTIL GET\$=" "

1790 ENDPROC

1800 REM

1810 REM Spool a character onto cassette

1820 REM

1830 DEF PROC SPOOL

1840 INPUT TAB(5,5)"First character number = "N

1850 IF N<32 OR N>255 OR N<>INT(N) THEN CLS:GOTO 1840

1860 INPUT TAB(5,10)"First line number = "L

1870 IF L<0 OR L>32767 OR L<>INT(L) THEN 1860

1880 *SPOOL("CHAR")

1890 FOR IX=0 TO D-1

1900 FOR JX=0 TO W-1

1910 PRINT ;L;" VDU 23,"N;

1920 FOR KX=0 TO 7

1930 PRINT "","? (TX+KX+JX*8+IX*80)

1940 NEXT KX

1950 PRINT

1960 N=N+1

1970 L=L+10

1980 NEXT JX,IX

1990 *SPOOL

2000 PROCSPACE(31)

2010 ENDPROC

2020 REM

2030 REM Invert the picture in memory. i.e. change all white blocks to black and visa-versa

2040 REM

2050 DEF PROC INVERT

2060 PRINT "Please wait a few seconds."

2070 FOR IX=0 TO YN*80+80

2080 MX=IX*(MX*IX+1)AND1

2090 NEXT IX

2100 ENDPROC

2110 REM

2120 REM List the binary codes making each character up

2130 REM

2140 DEF PROC LIST

2150 FOR IX=0 TO D-1

2160 FOR JX=0 TO W-1

2170 CLS

2180 PRINT "Character at ";JX*1,"";IX*1

2190 PRINT ""Codes are as follows

2200 FOR KX=0 TO 7

2210 PRINT TAB(20);"(TX+KX+JX*8+IX*80)

2220 NEXT KX

2230 PROCSPACE(31)

2240 NEXT JX,IX

2250 ENDPROC

2260 REM

2270 REM Allow you to make an O.S. call from within the program

2280 REM

2290 DEF PROC OSCALL

2300 PRINT TAB(0,5);"Type the required operating system call and press RETURN."

2310 PRINT TAB(0,10);"Call = "

2320 INPUT ""C\$

2330 \$%A0=C\$

2340 X%0=Y%=&A

2350 PRINT TAB(0,15);

2360 CALL &FFF7

2370 ENDPROC

A Fill routine

Douglas Stewart of Edinburgh has submitted another recursive procedure demonstrating that not all recursion is superfluous extravagance coded for effect rather than utility. This routine will fill an area of background colour bounded by non-background colour with foreground colour.

Within this bald description lies a process which is a joy to watch on the screen, as colour first flows up, then down, filling nooks and crannies of irregular shapes, and back-tracking to finish off part-completed sections. The Fill function is essential to any graphics art pack, and the test bed in which the procedure is set will allow pictures to be drawn. It provides a good grounding from which you can develop an art pack tailored to your own specification. The routine can be included within any program requiring a Fill facility; the testbed is optional.

310 DEF PROC delay(A\$):TX=TIME:REPEAT UNTIL TIME>TX+A\$:ENDPROC

320 DEF PROC orient:orient=ASN((y-edgex)/(FNhype+U.001)):IF x-edgex<0 THEN orient=orient+PI

330 ENDPROC

340 DEF PROC sides:PROC Gk("3456789abcdefghijklmnopqrstuvwxyz"):sides=ASC(a\$):IF sides>96 THEN sides=sides-86 ELSE sides=sides-48

350 ENDPROC

360 DEF FNhype:=SQRT((x-edgex)^2+(y-edgex)^2)

370 DEF PROC text:REPEAT:a\$=GET\$:PRINT a\$;UNTIL a\$=CHR\$(13):ENDPROC

38000 REM Procedure to fill an area of current background colour which is enclosed by non-background colour.

39020 REM Syntax is PROC FILL(Xcoordinate,Ycoordinate,current_background_colour)

39030 REM Procedure will work in any mode.

39040 REM By Douglas Stewart, March 1983.

39050 REM ***** NOTE: OPERATING SYSTEM SERIES 1 OR LATER ONLY *****

39060 REM -----

39070 REM As the function is recursive for more complex shapes, short variable

39080 REM names have been used to limit the stack space used.

39090 DEF PROC FILL(X,Y,V)

39100 DIM PARAM 7:REM Space for parameter block for OSW0KD 13.

39110 V=V AND 15

39120 LOCAL I

39130 AX=135:M=((USR(&FFF4)DIV 10000)AND 15)=1:REM Current graphics mode.

39140 IF M=7 OR M=3 OR M=6 THEN PROC:REM Check for non-graphics mode.

39150 W=2*(N MOD 3+1):REM Width of pixel for this mode.

39160 Z=2*W

39170 PROCUD(X,Y,4):PROCUD(X,Y,-4):REM FILL UP AND DOWN

39180 ENDPROC

39190 DEF PROCUD(X,Y,S)

39200 LOCAL FX,BX,CX,DX,EX:REM These variables must be LOCAL

39210 PL0T 6,X,Y:REM Get width but do not draw yet.

39220 BX=FN(4):REM Get last X coordinate.

39230 CX=FN(0):REM Get previous X coordinate.

39240 :

39250 REM ***** Main loop starts here. *****

39260 PL0T 7,X,Y:REM Fill in a horizontal line.

(continued on page 147)

Face

Alistair Lindsay of Edinburgh has submitted what he calls his first successful attempt at making a reasonable program. The result is a Mode 7 graphic display — not the easiest problem to start on! — which does just what it sets out to do; it is quite funny.

I can see a whole set of computer jokes resulting from it — variations on the "Mummy, Mummy" or "Knock, Knock" themes of yesteryear — but this is the first. Nobody has submitted a computerised joke before. Given enough of them chained together, it might be the music-hall turn reborn.

Box spin

R A Lober of Peterson-super-Ely, Cardiff, has submitted a demonstration of graphics rotation. I had not previously thought it possible in Basic but Mr Lober proves me wrong — though I suspect that is only because the box he is spinning is rather small and only four of its faces are ever visible. The following variables are used:

A% — width of sides
B% — width of ends
T% — angle of turn
D — current width of sides
P — current width of ends
S — perspective
W — reduction in D
F — reduction in P

```
(listing continued from page 143)
30270 Y=Y+S
30280 DX=FNC(4)
30290 EX=FNC(0)
30300 IFEX<CX<Z THEN30350:REM Extrem
ities almost coincident?
30310 FX=CX
30320 FX=FX+W
30330 IFPOINT(FX,Y-S)=V PROCUD(FX,Y-
S,S):REM Recurse to FILL branch
30340 IFFX<EXTHEN30320
30350 IFBX<DX<Z THEN30400
30360 FX=DX
30370 FX=FX+W
30380 IFPOINT(FX,Y-S)=V PROCUD(FX,Y-
S,S):REM Recurse to FILL branch.
30390 IFFX<EXTHEN30370
30400 IFEX<CX<Z THEN30450
30410 FX=EX
30420 FX=FX+W
30430 IFPOINT(FX,Y-2*S)=V PROCUD(FX,
Y-2*S,-S):REM Recurse to FILL branch
30440 IFFX<EXTHEN30420
30450 IFDX<BX<Z THEN30500
30460 FX=BX
30470 FX=FX+W
30480 IFPOINT(FX,Y-2*S)=V PROCUD(FX,
Y-2*S,-S):REM Recurse to FILL branch
30490 IFFX<EXTHEN30470
30500 EX=DX
30510 CX=EX
30520 IFPOINT(X,Y)<>V THEN30530ELSE3
0560
30530 IFPOINT(X,Y)=1ENDPROC:IE If i
t is off the edge of screen.
30540 FX=EX
30550 REPEAT
30560 FX=FX+W
30570 UNTILFX>DXORPOINT(FX,Y)=V
30580 IFFX>DXENDPROC
30590 X=FX
30600 GOTO30260
30610 :
30620 REM THIS FUNCTION USES OSWORD
13 TO GET THE LAST POINTS VISITED
30630 REM VARIABLE 0 IS THE OFFSET O
F THE AREA TO BE READ IN THE PARAM
BLOCK.
30640 DEFFNC(0):AX=13:XX=PARAM MOD25
5:YX=PARAM DIV255:CALL&FFF1:=(PARAM
0)AND&FFFF
```

Face

```
10 MODE7
20 PRINT CHR$(141);CHR$(129);CHR$(1
57);CHR$(135);"WELCOME TO THE B B C MIC
RO COMPUTER"
30 PRINT CHR$(141);CHR$(129);CHR$(1
57);CHR$(135);"WELCOME TO THE B B C MIC
RO COMPUTER"
40 PRINT CHR$(141);CHR$(129);CHR$(1
57);CHR$(135);"I AM GOING TO DEMONSTRAI
T WHAT I"
50 PRINT CHR$(141);CHR$(129);CHR$(1
57);CHR$(135);"I AM GOING TO DEMONSTRAI
T WHAT I"
60 PRINT CHR$(141);CHR$(129);CHR$(1
57);CHR$(135);" CAN DO"
70 PRINT CHR$(141);CHR$(129);CHR$(1
57);CHR$(135);" CAN DO"
80 PRINT:PRINT:PRINT:PRINT:PRINT CH
R$(133);" PRESS 'Y' TO GO ON OR 'N' TO GO
BACK"
90 AS=GET$
100 IF AS="N" THEN GOTO10
104 IF AS="Y" THEN GOTO110
105 GOTO90
110 CLS
120 MODE2
130 VDU23,88,255,255,255,255,255,255
,255,255:VDU23,81,254,254,254,126,62,30
,14,6,23,90,255,255,255,255,0,0,0,0
140 COLOUR3:PRINT "XXXXXX"
150 COLOUR3:PRINT "XXXXXXXXXX"
160 COLOUR3:PRINT "XXXXXXXXXXXX"
170 COLOUR3:PRINT "XX";:COLOUR5:P
RINT"XXXXXXXXXX";:COLOUR3:PRINT"XX"
180 COLOUR3:PRINT "X";:COLOUR5:PR
INT"XXXXXXXXXX";:COLOUR3:PRINT"X"
190 COLOUR5:PRINT "XXXXXXXXXXXXXX"
200 COLOUR5:PRINT "XXX";:COLOUR7:PR
INT"XXX";:COLOUR5:PRINT"X";:COLOUR1:PR
INT"XX";:COLOUR5:PRINT"X";:COLOUR7:PR
INT"XXX";:COLOUR5:PRINT"XXX"
210 COLOUR5:PRINT "XXX";:COLOUR7:PR
INT"XX";:COLOUR4:PRINT"X";:COLOUR5:PRIN
X";:COLOUR4:PRINT"X";:COLOUR7:PRINT"XX"
220 COLOUR5:PRINT "XXX";:COLOUR
1:PRINT"X";:COLOUR5:PRINT"XXXXXXXX"
230 COLOUR5:PRINT "XXXXXXXX";:COLOUR
1:PRINT"XX";:COLOUR5:PRINT"XXXXXXXX"
240 COLOUR5:PRINT "XXXX";:COLOUR1
:PRINT"XXXX";:COLOUR5:PRINT"XXXX"
250 COLOUR5:PRINT "XXXX";:COLOUR1
:PRINT"XXXX";:COLOUR5:PRINT"XXXX"
260 COLOUR5:PRINT "XXXXXXXXXXXXXX"
270 COLOUR5:PRINT "XXXXXXXXXXXX"
280 COLOUR5:PRINT "XX";:COLOUR1:
PRINT"XXXXXXXX";:COLOUR5:PRINT"XX"
290 COLOUR5:PRINT "X";:COLOUR1:
PRINT"X";:COLOUR0:PRINT"X";:COLOUR7:PR
INT"QZ";:COLOUR0:PRINT"X";:COLOUR1:PRIN
T"X";:COLOUR5:PRINT"X"
300 COLOUR5:PRINT "X";:COLOUR1:
PRINT"XXXXXXXX";:COLOUR5:PRINT"X"
310 COLOUR5:PRINT "XXXXXXXXXX"
320 COLOUR5:PRINT "XXXXXX"
330 PRINT:PRINT "I'VE GONE BANANAS"
340 FOR I=1 TO 3
350 COLOUR7:PRINTTAB(5,6);"XX";:COLO
UR4:PRINT"X";:COLOUR7:PRINTTAB(5,7);"X
XX";:COLOUR4:PRINTTAB(12,6);"X";:COLOUR7
:PRINT"XX";:COLOUR7:PRINTTAB(12,7);"XXX"
360 FOR J=1 TO 300:NEXTJ
370 COLOUR7:PRINTTAB(5,6);"X";:COLO
UR4:PRINT"X";:COLOUR7:PRINT"X";:COLOUR7
:PRINTTAB(5,7);"XXX";:COLOUR7:PRINTTAB(1
2,6);"X";:COLOUR4:PRINT"X";:COLOUR7:PRIN
T"X";:COLOUR7:PRINTTAB(12,7);"XXX"
380 FOR J=1 TO 300:NEXTJ
390 COLOUR4:PRINTTAB(5,6);"X";:COLO
UR7:PRINT"XX";:COLOUR7:PRINTTAB(5,7);"XXX
";:COLOUR7:PRINTTAB(12,6);"XX";:COLOUR4:
PRINT"X";:COLOUR7:PRINTTAB(12,7);"XXX"
400 FOR J=1 TO 300:NEXTJ
410 COLOUR7:PRINTTAB(5,6);"XXX";:COLO
UR4:PRINTTAB(5,7);"X";:COLOUR7:PRINT"XX
";:COLOUR7:PRINTTAB(12,6);"XXX";:COLOUR7:
PRINTTAB(12,7);"XX";:COLOUR4:PRINT"X"
420 FOR J=1 TO 300:NEXTJ
430 COLOUR7:PRINTTAB(5,6);"XXX";:COLO
UR7:PRINTTAB(5,7);"X";:COLOUR4:PRINT"X"
";:COLOUR7:PRINT"X";:COLOUR7:PRINTTAB(12
,6);"XXX";:COLOUR7:PRINTTAB(12,7);"X";:CO
LOUR4:PRINT"X";:COLOUR7:PRINT"X"
440 FOR J=1 TO 300:NEXTJ
450 COLOUR7:PRINTTAB(5,6);"XXX";:COLO
UR7:PRINTTAB(5,7);"XX";:COLOUR4:PRINT"X
";:COLOUR7:PRINTTAB(12,6);"XXX";:COLOUR4:
PRINTTAB(12,7);"X";:COLOUR7:PRINT"XX"
460 VDU31,0,0
461 NEXTI
470 FOR Z=0 TO 255
480 SOUND1,-15,Z,1
490 NEXTZ
491 FOR Y=155 TO 0 STEP-1
492 SOUND1,-15,Y,1
```

```
493 NEXTY
510 CS=INKEY$(10)
520 MODE7
530 PRINTCHR$(141);CHR$(136);CHR$(13
4);" HOW DID YOU LIKE THAT? IF YOU "
540 PRINTCHR$(141);CHR$(136);CHR$(13
4);" HOW DID YOU LIKE THAT? IF YOU "
550 PRINTCHR$(141);CHR$(136);CHR$(13
4);" DID PRESS 'Y' IF NOT PRESS 'N' "
560 PRINTCHR$(141);CHR$(136);CHR$(13
4);" DID PRESS 'Y' IF NOT PRESS 'N' "
590 DS=GET$
600 IF DS="Y" THEN GOTO630
610 IF DS="N" THEN GOTO 620
615 GOTO 600
620 PRINTCHR$(135);CHR$(157);CHR$(12
9);" YOU'VE HURT MY FEELINGS"
625 GS=INKEY$(1000)
626 GOTO10
630 CLS:PRINTCHR$(141);CHR$(136);CHR
$(133);CHR$(157);CHR$(131);"THANK YOU F
OR PRAISING MY ART FAN"
640 PRINTCHR$(141);CHR$(136);CHR$(13
3);CHR$(157);CHR$(131);"THANK YOU FOR P
RAISING MY ART FAN"
650 PRINTCHR$(141);CHR$(136);CHR$(1
33);CHR$(157);CHR$(131);" THATS
ALL FOLKS"
660 PRINTCHR$(141);CHR$(136);CHR$(1
33);CHR$(157);CHR$(131);" THATS
ALL FOLKS"
```

Box spin.

```
1 AX=0
10 REM BOX SPIN
20 REM (c) Copyright R.A.Lober
30 REM 2.3.83
31 AX=AX+1:IFAX<3 GOTO10
40 MODE0
50 AX=100
60 BX=50
70 O=50
80 VDU29,640,512;
90 TIME=0
100 FORTX=0T0360STEP10
110 D=AX*COS(RAD(TX))
120 P=BX*SIN(RAD(TX))
130 S=2*P/3
140 W=ABS(S/4)
150 F=D/25
160 GCOL0,1:MOVE128,70:DRAW128,180
170 GCOL4,1
180 FORCX=0T01
190 IFF<OGOTO210
200 PROCENDA
210 IFF<W<W-D GOTO230
220 PROCIDEA
221 FORDE=1T050:NEXT
230 IFF<OGOTO250
240 PROCENDB
250 IFF<W>-(D+W)GOTO270
260 PROCIDEB
270 :
280 NEXT:NEXT
290 RUN
300 DEFPROCIDEA
310 MOVE128-D-P-W,150+S+F
320 DRAW128-D-P-W,150+S+F
330 DRAW128-D-P-W,100+S-F
340 DRAW128-D-P-W,100-S-F
350 DRAW128-D-P-W,150+S+F
360 IFCX<OFORN=0T00:NEXT
370 ENDPROC
380 DEFPROCENDB
390 MOVE128-D-P-W,100-S-F
400 DRAW128-D-P-W,100-S-F
410 DRAW128-D-P-W,150+S-F
420 DRAW128-D-P-W,150+S+F
430 DRAW128-D-P-W,100-S-F
440 IFCX<OFORN=0T00:NEXT
450 ENDPROC
460 DEFPROCIDEB
470 MOVE128-D-P-W,100+S+F
480 DRAW128-D-P-W,100+S+F
490 DRAW128-D-P-W,150+S-F
500 DRAW128-D-P-W,150-S-F
510 DRAW128-D-P-W,100+S+F
520 IFCX<OFORN=0T00:NEXT
530 ENDPROC
540 DEFPROCENDB
550 MOVE128-D-P-W,100+S+F
560 DRAW128-D-P-W,100+S-F
570 DRAW128-D-P-W,150-S-F
580 DRAW128-D-P-W,150-S-F
590 DRAW128-D-P-W,100+S+F
600 IFCX<OFORN=0T00:NEXT
610 ENDPROC
```

APPLE PIE

by John Harris



Patch

AN EXCELLENT 16-sector disc-patch program or catalogue analyser is submitted by Mark Benson of Tring. Any sector of a regularly formatted Apple floppy — which excludes tiresome commercially screwed discs like VisiCalc — may be read, amended, written or rewritten, with comprehensive prompts at the appropriate point.

While it provides a perfect way of learning about catalogue and program layout if the Write options are avoided, it is a guaranteed way of losing disc data if you do not know what you are about. Given that you do, Mark Benson has written an excellent utility to do it with.

The associated assembler routine — for which I have no source code — is to be saved as:

BSAVE BPATCH,A&0295,L\$0113

For use on 13-sector discs the following amendments are needed:

Line 140. *16 becomes *13
Line 670. <16 becomes <13
Line 710. = 15 becomes = 12
Line 1050. = 15 becomes = 12

Patch.

*0295.0387

```
0295- A5 3C 29
0298- F8 85 3C 38 A5 3E E5 3C
02A0- A5 3F E5 3D B0 01 60 A5
02AB- 3C 20 DA FD A2 02 20 4A
02B0- F9 A0 00 B1 3C 20 DA FD
02B8- A2 01 20 4A F9 C8 0C 08
02C0- D0 F1 F0 02 00 00 A2 01
02C8- 20 4A F9 A0 00 B1 3C 09
02D0- B0 C9 A0 B0 04 A9 A0 B0
02D8- 06 C9 E0 90 02 E9 20 20
02E0- ED FD C8 0C 08 D0 E6 A9
02E8- 8D 20 ED FD 18 A5 3C 85
02F0- 40 69 08 85 3C A5 3D 85
02F8- 41 69 00 85 3D 90 9C 60
0300- 18 90 68 A9 04 D0 27 A5
0308- 1F 4C DA FD A9 01 D0 02
0310- A9 02 48 20 E3 03 85 49
0318- B4 48 A0 0C 68 91 48 20
0320- E3 03 20 D9 03 B0 06 A9
0328- 00 A0 0D 91 48 60 48 20
0330- E3 03 84 48 85 49 A0 04
0338- B1 48 85 1E C8 B1 48 85
0340- 1F 68 85 49 A9 24 85 48
0348- A2 02 A0 03 18 B5 1D 29
0350- 0F 08 18 69 B0 C9 BA 90
0358- 02 69 06 91 48 88 28 B0
0360- 09 85 1D 4A 4A 4A 4A 38
0368- B0 E7 CA D0 DF 60 20 E3
0370- 03 8C C0 03 8D C1 03 A9
0378- B6 85 48 AD E5 03 85 49
0380- A0 00 B1 48 D0 0D A5 CC
0388- 8D C2 03 A5 CD A6 CA A4
0390- CB 90 08 A5 6D 8D C2 03
0398- A5 6E A6 6F A4 70 8D C3
03A0- 03 8E C4 03 8C C5 03 60
```

Patch.

```
10 IF GO THEN 170
20 GO = 1
30 KB = - 16384:KS = - 16368:MD
= 661
40 TEXT: CALL - 936
50 PRINT CHR$(4):"BLOAD BPATCH"
60 PRINT "16 SECTOR DISC PATCH:"
70 PRINT
80 POKE 34,5
90 GOSUB 1450
100 B = PEEK (LP + 1) + 2
110 BB = B * 128:BB = BB + BB
120 S = PEEK (SL) / 16:D = PEEK
(DR)
130 PRINT "DISC": GOSUB 1300
140 POKE SL,S * 16: POKE DR,D: POKE
VL,0
150 POKE BL,0: POKE BH,B
160 GOSUB 1010:DS = 0: GOTO 550
170 POKE 36,0: POKE 37,22: CALL
- 990
180 CALL - 868
190 PRINT
200 PRINT " OPTION (TYPE ?
FOR MENU) ?":
210 POKE 50,63: PRINT " ": POKE
50,255
220 CALL - 868
230 POKE 36,32
240 CH = PEEK (KB): IF CH < 128 THEN
240
250 POKE KS,0: POKE 36,0: CALL -
868
260 IF CH = 212 THEN 380
270 IF CH = 194 THEN 390
280 IF CH = 208 THEN 470
290 IF CH = 210 THEN 550
300 IF CH = 215 THEN 610
310 IF CH = 171 THEN 670
320 IF CH = 173 THEN 700
330 IF CH = 206 THEN 160
340 IF CH = 195 THEN 780
350 IF CH = 191 THEN 800
360 IF CH = 155 THEN 1000
370 CALL - 198: GOTO 170
380 DS = 0: GOTO 400
390 DS = 128
400 POKE 36,0: POKE 37,5: CALL -
990
410 CALL - 868
420 POKE 37,6: CALL - 990
430 POKE 60,DS: POKE 62,DS + 127
440 POKE 61,B: POKE 63,B
450 CALL MD
460 GOTO 170
470 POKE 36,0: POKE 37,4: CALL -
990
480 CALL - 868
490 PRINT "ADDR: ":
500 MX = 255: GOSUB 1130: IF TS <
0 THEN 170
510 AD = TS: PRINT " PATCH: ":
520 GOSUB 1130: IF TS < 0 THEN 4
00
530 POKE BB + AD,TS:AD = AD + 1:
IF AD > 255 THEN 400
540 PRINT " ": GOTO 520
550 POKE 36,0: POKE 37,4: CALL -
990
560 CALL - 868
570 POKE PF,210: CALL DT
580 CALL RD
590 IF PEEK (RP) < > 0 THEN GOSUB
1550
600 GOTO 400
610 POKE 36,0: POKE 37,4: CALL -
990
620 CALL - 868
630 POKE PF,215: CALL DT
640 CALL WR
650 IF PEEK (RP) < > 0 THEN GOSUB
1550
660 GOTO 170
670 S = S + 1: IF S < 16 THEN 730
680 S = 0:T = T + 1: IF T < 35 THEN
730
690 T = 0: GOTO 730
700 S = S - 1: IF S > = 0 THEN 7
30
710 S = 15:T = T - 1: IF T > = 0
THEN 730
720 T = 34
730 DS = 0
740 POKE 37,2: CALL - 990
750 POKE 36,26: POKE HX,T: CALL
PX
760 POKE 36,36: POKE HX,B: CALL
PX
770 POKE TK,T: POKE SE,S: GOTO 5
50
780 GOSUB 1010
790 GOTO 400
```

```
800 POKE 36,0: POKE 37,6: CALL -
990
810 POKE 35,22: CALL - 936
820 POKE 35,24
830 PRINT " OPTIONS AVAILABLE
": PRINT
840 POKE 32,5: POKE 33,35: CALL
- 990
850 PRINT "T DISPLAY BYTES 00-7
F.":
860 PRINT "B DISPLAY BYTES 80-F
F.": PRINT
870 PRINT "R RE-READ CURRENT SE
CTOR.":
880 PRINT "W WRITE BACK CURRENT
SECTOR.": PRINT
890 PRINT "+ ADVANCE A SECTOR."
900 PRINT "- GO BACK A SECTOR."
: PRINT
910 PRINT "N SPECIFY A NEW SECT
OR.":
920 PRINT "C SPECIFY A NEW SECT
OR BUT"
930 PRINT " DO NOT READ IT IN.
": PRINT
940 PRINT "P PATCH THE CURRENT
SECTOR"
950 PRINT " IN MEMORY."
960 POKE 32, PEEK (32) - 1: CALL
- 990
970 PRINT "ESC END."
980 POKE 32,0: POKE 33,40: CALL
- 990
990 GOTO 170
1000 TEXT: END
1010 POKE 36,19: POKE 37,2: CALL
- 990
1020 PRINT " TRACK ":MX = 34: GOSUB
1130
1030 IF TS < 0 THEN 1010
1040 T = TS: POKE TK,T
1050 PRINT " SECTOR ":MX = 15: GOSUB
1130
1060 IF TS < 0 THEN 1010
1070 S = TS: POKE SE,S
1080 POKE 36,0: POKE 37,3: CALL
- 990
1090 CALL - 868
1100 POKE 37,4: CALL - 990
1110 CALL - 868
1120 RETURN
1130 CALL - 868
1140 CH = PEEK (36):CV = PEEK (
37): POKE 51,128
1150 CALL - 662
1160 POKE 36,CH: POKE 37,CV: CALL
- 990
1170 PT = 512:CH = PEEK (PT):TS =
- 1: IF CH = 141 THEN 1280
1180 TS = 0
1190 IF CH = 131 THEN END
1200 CH = CH - 176
1210 IF CH < 0 THEN 1290
1220 IF CH > 22 THEN 1290
1230 IF CH < 10 THEN 1250
1240 CH = CH - 7: IF CH < 10 THEN
1290
1250 TS = TS + 16 + CH: IF TS > M
X THEN 1290
1260 PT = PT + 1:CH = PEEK (PT):
IF CH < > 141 THEN 1190
1270 POKE HX,TS: CALL PX
1280 CALL - 868: RETURN
1290 CALL - 198: GOTO 1130
1300 POKE - 16368,0: CALL - B6
8
1310 PRINT " SLOT ":SD = S:MX =
8: GOSUB 1340:SD = SD
1320 PRINT " DRIVE ":SD = D:MX =
3: GOSUB 1340:D = SD
1330 CALL - 868: RETURN
1340 CH = PEEK (36)
1350 POKE 50,63: PRINT SD: POKE
50,255
1360 POKE 36,CH
1370 CH = PEEK (- 16384): IF CH
< 128 THEN 1370
1380 POKE - 16368,0
1390 IF CH = 141 THEN 1440
1400 CH = CH - 176
1410 IF CH > 0 AND CH < MX THEN
1430
1420 CALL - 198: GOTO 1370
1430 SD = CH
1440 PRINT SD: RETURN
1450 GI = 768:DT = 771:PX = 775:R
D = 780:WR = 784:IP = 960:LP
= 962:HP = 964
1460 CALL GI
1470 I = IP: GOSUB 1530:IB = J
1480 I = LP: GOSUB 1530:LO = J
1490 I = HP: GOSUB 1530:HI = J
1500 SL = IB + 1:DR = IB + 2:VL =
IB + 3:TK = IB + 4:SE = IB +
5:BL = IB + 8:BH = IB + 9:RP
= IB + 13
1510 HX = 31:FF = 1057
```

(continued on next page)

Sub exterminator.

*5600.5920

```

5600- 07 00 10 00 03 01 F4 01
5608- 70 02 F5 02 09 03 1D 03
5610- 4B 49 49 49 49 49 29 2D
5618- 2D F5 DB 1B 4D 49 DE DB
5620- 2B 2D 2D F5 DB 1B 2D 2D
5628- 2D DE DB 2B 2D 2D 6D 49
5630- 49 49 49 49 29 DE DB DB
5638- DB DB DB DB DB DB DB DB
5640- DB 1B 4D 49 0D 2D 2D 2D
5648- 2D 2D 2D 2D 2D 2D 2D 2D
5650- 2D 2D 2D F5 DB DB DB DB
5658- DB DB DB DB DB DB DB DB
5660- 0D 2D 0D 0D 2D 2D 2D 2D
5668- 2D 2D 2D 2D 2D 4D 49 4D
5670- 29 DE DB DB DB DB DB DB
5678- DB DB DB DB DB DB DB 2D
5680- 2D 2D 2D 2D 6D 49 49 49
5688- 49 49 49 4D F1 DB DB DB
5690- DB DB DB DB DB DB DB DB
5698- DB DB 2D 2D 2D 2D 6D 49
56A0- 49 49 4D 09 4D 09 4D 09
56A8- 2D 0D F5 DB DB DB DB DB
56B0- DB DB DB DB DB DB DB DB
56B8- 4D 49 49 49 49 49 49 49
56C0- 49 49 49 49 4D DE DB DB
56C8- DB DB DB DB DB DB DB DB
56D0- DB DB 6B 49 49 49 49 49
56D8- 49 49 49 49 49 49 F5
56E0- DB DB DB DB DB DB DB DB
56E8- DB DB DB DB 1B 2D 2D 2D
56F0- 2D 2D 2D 2D 2D 2D 2D 2D
56F8- 2D 2D 2D 2D 2D 2D 2D 2D
5700- 4E 09 00 4B 49 49 49 49
5708- 49 09 2D 2D 2D DE DB 6B
5710- 49 F1 DB 1B 2D 2D 2D DE
5718- DB 2B 2D 2D F5 DB DB DB
5720- DB DB DB DB 1B 6D 49 49
5728- 49 49 49 2D 2D 2D DE DB
5730- DB DB DB DB DB DB 2B 2D
5738- 2D 2D 2D 2D 2D 2D 2D 2D
5740- 2D 2D 2D 2D 6D 49 49 F1
5748- DB DB DB DB DB DB DB DB
5750- DB DB DB DB 6D 09 4D 49
5758- 2D 2D 2D 2D 2D 2D 2D 2D
5760- 6D 69 2D 0D DE DB DB DB
5768- DB DB DB DB DB DB DB DB
5770- 6B 09 4D 49 49 49 49 49
5778- 09 2D 2D 2D 2D 2D F5
5780- DB DB DB DB DB DB DB DB
5788- DB DB DB DB 6D 2D 4D 09
5790- 4D 09 4D 09 4D 49 49 09
5798- 2D 2D 2D 2D F5 DB DB DB
57A0- DB DB DB DB DB DB DB DB
57A8- DB 1B 4D 4D 49 49 49 49
57B0- 49 49 49 49 49 49 DE
57B8- DB DB DB DB DB DB DB DB
57C0- DB DB DB DB 2B 4D 49 49
57C8- 49 49 49 49 49 49 49 49
57D0- 49 09 DE DB DB DB DB DB
57D8- DB DB DB DB DB DB DB 2D
57E0- 2D 2D 2D 2D 2D 2D 2D 2D
57E8- 2D 2D 0E 00 4B 49 49 29
57F0- 2D 2D 0E 00 4B 49 49 29
57F8- 1E F5 F5 DB 2D 2D 2D 2D
5800- 6D 69 09 4D F1 DB DB DB
5808- DB DB DB DB 2B 2D 2D 2D
5810- 2D 2D 2D 2D 2D 2D 2D 2D
5818- 2D F5 DB DB DB DB DB DB
5820- DB DB 1B 6D 49 49 0D 4D
5828- 49 0D 4D 49 49 DE DB DB
5830- DB DB DB DB DB DB DB 2D
5838- 4D 49 09 0D 4D 4D 0D 4D
5840- 49 49 F5 DB DB DB DB DB
5848- DB DB DB DB 2B 4D 49 09
5850- 2D 4D 49 0D 4D 49 49 DE
5858- DB DB DB DB DB DB DB DB
5860- 2B 2D 2D 2D 2D 2D 2D 2D
5868- 2D 2D 2D 2D 2D 75 09 00
5870- 4B 49 49 49 49 49 49 49
5878- F5 F5 F5 DB DB DB DB DB
5880- 6B 09 4D 69 29 DB 2D 2D
5888- 2D DE DB DB DB DB DB DB
5890- DB 2D 2D 2D 2D 2D 2D 2D
5898- 2D 2D 2D 2D 2D 2D DE DB
58A0- DB DB DB DB DB DB DB 6B
58A8- 49 49 69 4D 49 2D 4D 49
58B0- 09 F5 DB DB DB DB DB DB
58B8- DB DB DB DB 4D 49 49 0D
58C0- 4D 4D 0D 4D 49 09 2D DE
58C8- DB DB DB DB DB DB DB DB
58D0- DB 4D 49 49 0D 4D 49 0D
58D8- 4D 49 09 F5 DB DB DB DB
58E0- DB DB DB DB 1B 2D 2D 2D
58E8- 2D 2D 2D 2D 2D 2D 2D 2D
58F0- 2D 2D 4E 01 00 4B F1 6B
58F8- 0D DE 2B F5 1B 2D 2D DE
5900- 2B F5 1B 0D 0D DE 4E 01
5908- 00 4B 09 1E 2D DE 2D DE
5910- 2D DE 2D DE 2B 2D F5 DB
5918- 6B 0D 0D 0E 00 FF 00 00
5920- 00
*3D0B

```

(continued from previous page)

```

1520 RETURN
1530 J = PEEK (I + 1): IF J > 12
7 THEN J = J - 256
1540 J = J * 128: J = J + J + PEEK
(I): RETURN
1550 PRINT "ERR":
1560 POKE HX, PEEK (RP): CALL PX
1570 PRINT " AT S": PEEK (SL) /
161 " D": PEEK (DR): " TRK ":
1580 POKE HX, PEEK (TK): CALL PX
1590 PRINT " SEC ":
1600 POKE HX, PEEK (SE): CALL PX
1610 PRINT
1620 RETURN

```

Sub exterminator

The graphics on this game from Mark Heather of Cudham are excellent, and the play is exciting once the controls have been mastered. Having tried for some time, both looking at the listing and running the program, I can still not deduce the submarine's strategy or quite how to stay afloat for any length of time, but that is what makes it so interesting. Mr Heather does not say what utility he used to generate his graphics elements, but they are quite superb.

Sub exterminator.

```

10 REM SUB-EXTERMINATOR
15 REM BY M.J.HEATHER
20 REM ON APPLE 2
25 REM 30/4/83
30 REM
35 IF PEEK (22016) = 7 AND PEEK
(22017) = 0 THEN GOTO 45
40 PRINT CHR$ (4): "BLOOD SUB SH
APES"
45 POKE 232,0: POKE 233,86
50 CLEAR: GOSUB 615: GR: TEXT
52 ONERR GOTO 5000
55 M$ = " *** YOU HIT ***
***YOU HIT ***
60 N$ = " *** YOU HIT
***
65 HOME: PRINT "INSTRUCTIONS:-"
: PRINT
70 PRINT: PRINT "YOU HAVE TO LI
NE YOUR BOAT WITH THE SUB"
75 PRINT "THEN LAUNCH A MINE, BU
T YOU ONLY HAVE"
80 PRINT "10 MINES !!!": PRINT
85 PRINT "THE SUB CAN ALSO BLOW
YOU OFF THE FACE"
90 PRINT "OF THE EARTH, SO BEWA
RE AND GOOD LUCK!": PRINT: PRINT
"CONTROLS:-": PRINT
95 PRINT "LEFT ARROW = MOVE LE
FT"
100 PRINT "RIGHT ARROW = MOVE R
IGHT"
105 PRINT "SPACE BAR = MINE
LAUNCH"
110 PRINT
115 PRINT "ANY OTHER KEY TO MOVE
SHIP TO RANDOM": PRINT "PO
SITION AND ANCHOR"
120 INVERSE: PRINT "WARNING YOU
LOSE A MINE AFTER DOING THI
S": NORMAL: PRINT
125 FLASH: PRINT "PRESS SPACE B
AR TO START": NORMAL
130 GET A$
135 REM START OF GAME
140 HGR: HCOLOR= 3: SCALE= 1: ROT=
0: W= 10: REM SET GRAPHIC
S AND NO. OF MINES
145 HOME
150 VTAB 21: PRINT "MINES = "; W
155 PRINT TAB( 30): "SCORE = "; S
C
160 S = INT ( RND (1) * 80) * 3 +
10
165 R = 3
170 IF S < SM THEN R = 4
175 E = PEEK ( - 16384): OX = X: VTAB
21: PRINT "MINES = "; W: PRINT
TAB( 30): "SCORE = "; SC: E =
E - 128: IF E = 8 THEN X = X
- 8
180 IF E = 21 THEN X = X + 8
185 IF E = 32 THEN GOTO 260
190 IF E > 32 THEN X = INT ( RND
(1) * 235 + 1): POKE - 1636
B, OX = W - 1: IF W = 0 THEN
GOTO 550
195 IF X < 1 THEN X = 1
200 IF X > 235 THEN X = 235
205 IF X > OX THEN Z = 1
210 IF X < OX THEN Z = 2
215 IF RND (1) > .85 THEN GOTO
400
220 DRAW Z AT X, 3
225 POKE PI, INT ( RND (1) * 30 +
100): POKE DU, 4: CALL NO: HPL0T
0, 15 TO 279, 15: IF SM < S THEN
D = SM + 5
230 IF SM > S THEN D = SM - 5
235 SM = D: DRAW R AT SM, 150: IF
SM > (S - 3) AND SM < (S + 3
) THEN HGR: GOTO 160
240 VTAB 21: PRINT "MINES = "; W:
: PRINT TAB( 30): "SCORE = "
: SC: XDRAW Z AT X, 3: XDRAW R
AT SM, 150: GOTO 175
245 REM
250 REM SHIP FIRING
255 REM
260 DRAW Z AT X, 3: XDRAW R AT SM
, 150: HPL0T X + 10, 5: FOR A =
10 TO 150 STEP 15: DRAW 5 AT
X + 10, A
265 POKE PI, SM: POKE DU, 3: CALL
NO
270 XDRAW R AT SM, 150
275 IF SM < S THEN D = SM + SC
280 IF SM > S THEN D = SM - SC
285 SM = D: DRAW R AT SM, 150: XDRAW
R AT SM, 150
290 IF SM > (S - 5) AND SM < (S +
5) THEN S = INT ( RND (1) *
86) * 3 + 10: DRAW Z AT X, 3:
HPL0T 0, 15 TO 279, 15: R = 3:
IF S < SM THEN R = 4: DRAW
R AT SM, 150
295 VTAB 21: PRINT "MINES = "; W:
: PRINT TAB( 30): "SCORE = "
: SC
300 XDRAW S AT X + 10, A
305 XDRAW R AT SM, 150
310 NEXT A
315 IF X + 10 > (SM - 1) AND X +
10 < (SM + 26) THEN 345
320 W = W - 1: IF W = 0 THEN GOTO
550
325 HGR: GOTO 175
330 REM
335 REM HIT SUB
340 REM
345 GR: HOME: SC = SC + 1
350 PRINT "YOU HIT THE SUB. SC0
RE = "; SC
355 FOR J = 1 TO 15 STEP 2: POKE
DU, 1: FOR C = 1 TO 3: F = INT
( RND (1) * 15) + 1: POKE PI
, F: CALL NO: NEXT C
360 COLOR= J: FOR H = 0 TO 39: POKE
PI, 40 - H: CALL NO: HLIN 0, 3
9 AT H: NEXT H: NEXT J
365 TEXT: HOME: FLASH: FOR V =
1 TO 24: POKE PI, V * 2: POKE
DU, 10: CALL NO: PRINT M$IN$:
: NEXT V: NORMAL
370 FOR I = 1 TO 20: POKE PI, 255
- I: CALL NO: POKE 32, 20 -
I: POKE 33, 2 * I: PRINT: PRINT
: NEXT: FOR I = 1 TO 24: POKE
PI, SC * 10: CALL NO: PRINT:
NEXT
375 X = INT ( RND (1) * 35) + 1
380 HGR: GOTO 175
385 REM
390 REM SHIP HIT
395 REM
400 DRAW R AT SM, 150: DRAW Z AT
X, 3
405 FOR A = 150 TO 3 STEP - 5: DRAW
6 AT SM + 10, A: XDRAW 6 AT S
M + 10, A: NEXT A
410 IF SM + 10 > X AND SM + 10 <
(X + 40) THEN GOTO 425
415 HGR
420 GOTO 175
425 TEXT: HOME: FOR A = 1 TO 1
2
430 "RINT "Y Y 000 U U 666 000
TTT H H III TTT !"
435 POKE PI, 20: POKE DU, 50: CALL
NO
440 PRINT "Y Y 000 U U 666 000
T H H I T !"
445 POKE PI, 45: CALL NO
450 PRINT "Y Y 000 U U 6 00
T H H I T !"
455 POKE PI, 50: CALL NO

```

HGR strings.

```

10 REM *** HGR STRINGS *** W.K
   .HO
20 DIM CA(23),CH(48,6)
30 ONERR GOTO 750
40 FOR I = 0 TO 5: READ C(I): NEXT
50 DATA 1,2,3,5,6,7
60 REM *** SET CORE ADDRESS FOR
   LINE
70 FOR I = 0 TO 23
80 READ CA(I): NEXT I
90 DATA 8192,8320,8448,8576,870
   4,8832,8960,9088
100 DATA 8232,8360,8488,8616,87
   44,8872,9000,9128
110 DATA 8272,8400,8528,8656,87
   84,8912,9040,9168
120 REM *** LOAD BIT PATTERN
130 FOR I = 0 TO 48
140 FOR K = 0 TO 6
150 READ CH(I,K): NEXT K
160 NEXT I
170 GOTO 390
180 DATA 0,0,8,28,8,0,0,0,0,8,8
   ,8,4,0,0,0,0,28,0,0,0
190 DATA 0,0,0,8,0,0,0,0,0,32,16,
   8,4,2,0
200 DATA 28,34,34,34,34,34,28,8
   ,8,8,8,8,8,28,34,16,8,4,2,
   62
210 DATA 28,34,32,24,32,34,28,2
   4,20,18,18,62,16,16,62,2,2,3
   0,32,34,28
220 DATA 28,34,2,30,34,34,28,62
   ,32,32,16,8,4,2,28,34,34,28,
   34,34,28
230 DATA 28,34,34,60,32,34,28,0
   ,8,8,0,8,8,0
240 DATA 0,8,8,0,8,8,4,0,16,8,4
   ,8,16,0,0,0,28,0,28,0,0
250 DATA 0,4,8,16,8,4,0,8,20,20
   ,16,8,0,8,8,20,28,20,12,20,8
260 DATA 8,28,34,34,62,34,34
270 DATA 30,34,34,30,34,34,30
280 DATA 28,50,2,2,2,50,28
290 DATA 30,34,34,34,34,34,30
300 DATA 62,2,2,30,2,2,62,62,2,
   2,30,2,2,2
310 DATA 28,34,2,2,58,34,28,34,
   34,34,62,34,34,34,8,8,8,8,
   8,8,32,32,32,32,34,28
320 DATA 50,10,6,6,10,26,50,2,2
   ,2,2,2,62,34,54,42,42,34,3
   4,34,34,38,42,42,42,50,34
330 DATA 28,34,34,34,34,34,28
340 DATA 30,34,34,30,2,2,2,28,3
   4,34,34,42,26,44,30,34,34,30
   ,10,18,50
350 DATA 28,34,2,28,32,34,28,62
   ,8,8,8,8,8,34,34,34,34,34,
   34,28
360 DATA 34,34,20,20,28,8,8,34,
   34,42,42,62,54,34,34,54,20,8
   ,20,54,34
370 DATA 34,34,20,28,8,8,8,62,3
   2,16,28,4,2,62
380 DATA 0,0,0,0,0,0,0,0
390 GOSUB 760: GOSUB 630: GOTO 7
   10
400 PRINT : PRINT : PRINT : PRINT

410 REM *** NOW READY TO WRITE
420 HGR : REM *** TEXT
430 PRINT : PRINT : PRINT
   "GIVE TEXT STRING- / CLR SCR
   "RET TO EXIT": INPUT S$
440 IF S$ < ">" GOTO 460
450 RETURN
460 IF S$ = "/" THEN HGR : GOTO
   430: REM ** "/" TO WIPE SCR
   EEN
470 K = INT ( RND (1) * 25): L =
   INT ( RND (1) * 20)
480 GOSUB 490: GOTO 430
490 N = LEN (S$): REM *** WRITE
500 IF N + K > 39 THEN N = 39 -
   K
510 FOR I = 1 TO N
520 SB$ = MID$(S$,I): REM ***
   READ EACH CHARACTER IN STRIN
   G
530 J = ASC (SB$): J = J - 43
540 IF J > -1 AND J < 48 GOTO
   570
550 J = 48
560 REM *** WRITE CHARACTER
570 FOR M = 0 TO 6
580 A = CA(L) + K + M * 1024 - 1
590 POKE A + 1,CH(J,M)
600 NEXT M
610 NEXT I
620 RETURN
630 TEXT : HOME : REM *** MENU
   VTAB 5
640 PRINT "HGR TEXT STRING GENER
   ATOR": PRINT : PRINT "BY W.K
   .HO-MAY,1983": PRINT : PRINT
660 PRINT "THIS PROGRAM GENERATE
   S TEXT STRINGS": PRINT "IN H
   GR MODE"
670 PRINT "THIS IS A DEMONSTRATI
   ON": PRINT : PRINT "OPTIONS:
   "1: PRINT : PRINT "0-MENU": PRINT
   "1-TEXT STRINGS:/TO CLEAR:RE
   T TO EXIT"
680 PRINT "2-SINE DEMO": PRINT "
   3-RANDOM WALK": PRINT "4-AUT
   O STRINGS": PRINT "5-AUTO SE
   QUENCE": PRINT "9-END"
690 VTAB 24
700 PRINT : RETURN
710 PRINT "CHOOSE OPTIONS-": PRINT
   "0-MENU:1-TEXT STR:2-SINE DE
   MO": PRINT "3-RANDOM WALK:4-
   AUTO TEXT:5-SEQ:9-EXIT": GET
   T: T = T + 1
720 IF T > 6 THEN END
730 ON T GOSUB 630,420,780,1040,
   1270,1310
740 GOTO 710
750 PRINT "ERROR-REENTER!": GOTO
   710
760 R$(0) = "START": R$(1) = "NO.1"
   : R$(2) = "NO.2": R$(3) = "NO
   .3": R$(4) = "NO.4"
770 R$(5) = "NO.5": R$(6) = "NO.6"
   : R$(7) = "NO.7": R$(8) = "END"
   : RETURN
780 N = INT ( RND (1) * 6)
790 IF (N > -1 AND N < 6) GOTO
   810
800 N = 0
810 HCOLOR= C(N)
820 PRINT : PRINT : PRINT "SINE-
   DEMO"
830 XO = 0
840 X = 0
850 Y = X
860 HGR
870 YO = XO
880 PI = 3.14159
890 HPLOT 4,80 TO XO,YO + 80
900 FOR I = 1 TO 80
910 X = X + 3
920 Y = 80 + YO + INT (40 * SIN
   (I * 0.1 * PI))
930 HPLOT TO X,Y
940 NEXT I
950 N = N + 1
960 IF (N < 5) GOTO 980
970 N = 0
980 HCOLOR= C(N)
990 X = XO + 6
1000 YO = X / 2
1010 XO = X
1020 IF (XO < 26) GOTO 890
1030 K = 10: L = 2: S$ = "SINE FUNC
   TION": GOSUB 490: RETURN
1040 HGR : REM *** RANDOM WALK
1050 N = INT ( RND (1) * 6)
1060 HCOLOR= C(N): FOR I = 70 TO
   90: HPLOT 100,I TO 120,I: NEXT
   I
1070 PRINT : PRINT : PRINT "RAND
   OM WALK"
1080 NN = INT ( RND (1) * 6): K =
   17: L = 11: S$ = "START": GOSUB
   490
1090 FOR P = 1 TO 7
1100 HCOLOR= C(NN)
1110 HPLOT 110,80
1120 XO = 110
1130 YO = 80
1140 FOR F = 1 TO 100
1150 X = XO + INT (19 * RND (1)
   ) - 9
1160 Y = YO + INT (19 * RND (1)
   ) - 9
1170 IF (X < 0) OR (X > 220) GOTO
   1230
1180 IF (Y < 0) OR (Y > 150) GOTO
   1230
1190 HPLOT TO X,Y
1200 XO = X
1210 YO = Y
1220 NEXT F
1230 NN = NN + 1
1240 IF (NN < 6) GOTO 1260
1250 NN = 0
1260 K = INT (XO / 7 + 1): L = INT
   (YO / 8 + 1): S$ = R$(P): GOSUB
   490: NEXT P: RETURN
1270 HGR : REM *** RANDOM TEXT
1280 VTAB 24: PRINT : PRINT : PRINT
   "RANDOM TEXT"
1290 FOR P = 1 TO 20: K = INT ( RND
   (1) * 25 + 1): L = INT ( RND
   (1) * 20 + 1): V = INT ( RND
   (1) * 9): S$ = R$(V): GOSUB 4
   90: NEXT P: RETURN
1300 FOR PS = 1 TO 5000: NEXT : RETURN
1310 GOSUB 630: GOSUB 1300: GOSUB
   780: GOSUB 1300: GOSUB 1040:
   GOSUB 1300: GOSUB 1270: GOSUB
   1300: RETURN

```

HGR strings

The Apple high-resolution graphics mode is limited by the lack of any dedicated character set for user applications such as captioning of the graphics display. This has resulted in more reinventing of wheels of so many shapes that I have long since lost count of how

many have come my way. However, this utility from Mr W K Ho of Cheltenham is particularly attractive and commands attention.

Though comparatively short it compiles a character set by specifying each character in a bit pattern occupying a seven-by-eight matrix which corresponds to the size of

each screen character in the Text mode. The starting addresses of each print line within HGR are also identified.

Various self-documented options are in-built to demonstrate different combinations of graphics and text. The appropriate areas of code may be included within user programs as desired. M

```

460 PRINT " Y O O U U 8 8 0 0
   T H H I T !"
465 POKE PI,55: CALL NO
470 PRINT " Y O O U U 666 0 0
   T H H I T !"
475 POKE PI,60: CALL NO
480 PRINT " Y O O U U 8 0 0
   T H H I T !"
485 POKE PI,65: CALL NO
490 PRINT " Y O O U U 8 0 0 0
   T H H I I T !"
495 POKE PI,70: CALL NO
500 NORMAL
505 IF A = 9 THEN HOME : FLASH
510 IF A = 10 THEN INVERSE
515 IF A < 9 THEN PRINT
520 NEXT A
525 PRINT "YOU LIVED TO SCORE "
   SC: " POINTS"
530 END
535 REM
540 REM GAME OVER
545 REM

550 FOR A = 1 TO 20: POKE DU,5:
   FOR B = 15 TO 1 STEP - 1: POKE
   PIT,B: CALL NOISE: NEXT B: NEXT
   A
555 FOR A = 1 TO 255: POKE DU,1
   : POKE PI,A: CALL NO: NEXT A
   : TEXT : HOME
560 FOR A = 1 TO 24
565 PRINT "GAME OVER":
570 INVERSE : PRINT " GAME OVER
   ": NORMAL
575 PRINT "GAME OVER":
580 FLASH : PRINT " GAME OVER ":
   : NORMAL
585 NEXT A
590 PRINT "YOU RAN OUT OF MINES,
   BUT SCORED "SC: " PTS."
595 END
600 REM
605 REM MUSIC LOCATIONS
610 REM
615 POKE 768,160: POKE 769,255: POKE
   770,162
620 POKE 771,160: POKE 772,202: POKE
   773,208
625 POKE 774,253: POKE 775,173: POKE
   776,48
630 POKE 777,192: POKE 778,136: POKE
   779,208
635 POKE 780,245: POKE 781,96
640 NO = 768: REM NOISE ROUTINE
645 PI = 771: DU = 769: REM PITCH
   & DURATION ROUTINES
650 RETURN
5000 FOR A = 1 TO 50
5010 POKE PI,A: POKE DU,250 - A:
   CALL NO
5020 NEXT A
5030 TEXT : HOME
5040 PRINT "BILLY": PRINT "====
   "
5050 PRINT : PRINT "(R)UN , (Q)U
   IT >>>?": SET P$
5060 IF P$ = "R" THEN CLEAR : GOTO
   40
5070 IF P$ = "Q" THEN HOME : END
5080 GOTO 5030

```


RESEARCH MACHINES REVIEW



Helicopter rescue

IN THIS program by Steve Crick of Herne Bay, Kent you are the pilot of a helicopter which must land on an oil rig to pick up survivors and take them over to the

waiting ship. Do not try to land on the oil rig if you already have survivors, as the overload will cause you to crash.

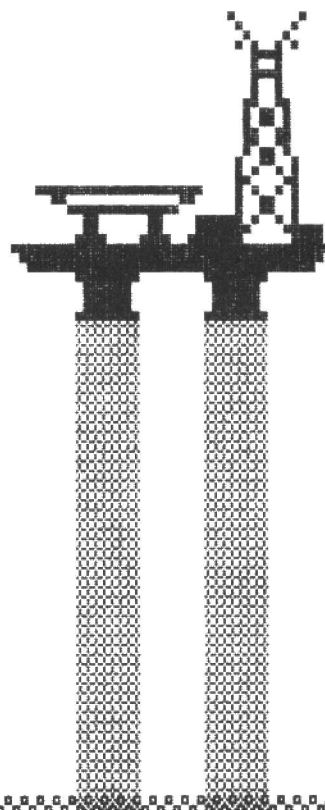
Winds can come from the left or right in forces from weak to gale force. At higher skill levels they can become stronger.

Because the computer has a read-ahead keyboard, if you keep hitting the direction you need to go in it will remember the sequence and carry it out. If you tap the Down key when landing, the buffer will remember Down and on take off you will crash. An auto repeat is incorporated into the game so the key need only be pressed once.

The machine code called at 750 works as a Get statement, but does so slightly faster.

Summary of listing.

10—130	Set up computer
140—470	Set user characters
480—730	Draw graphics
740—840	Move
850—970	Check position
980—1100	Random wind
1110—1230	Pick up survivors
1240—1380	Crash into rig
1390—1470	Land on ship with no survivors
1480—1620	Land on ship with survivors
1630—1750	Land in sea
1760—1900	Flown too high
1910—2160	Game over
2170—2390	Instructions
2400—2450	Lines cut out of main loop
2460—2530	Machine code
2540—2680	Skill level



Helicopter rescue

```

10 REM NORTH SEA OIL RIG RESCUE
20 REM Steve crick. May/83
30 REM Herne Bay Secondary School
40 REM For RML 480Z with level 2
50 REM High Resolution Graphics
60 RANDOMIZE
70 CALL "RESOLUTION",0,2
80 CLEAR 2000
90 GOSUB 2460
100 GOSUB 2170
110 GRAPH
120 ME=3
130 HS=1090
140 REM          SET CHARACTERS
150 CALL "DEFCHAR",1,0,0,0,252,92,127,59,31
160 CALL "DEFCHAR",2,0,0,0,0,255,183,255
170 CALL "DEFCHAR",3,14,30,63,22,30,255,222,252
180 CALL "DEFCHAR",4,63,0,64,127,85,95,1,15
190 CALL "DEFCHAR",5,252,192,220,226,225,209,78,240
200 CALL "DEFCHAR",6,16,8,4,10,3,2,3,2
210 CALL "DEFCHAR",7,8,16,32,80,192,64,192,64
220 CALL "DEFCHAR",8,2,6,5,5,6,6,3,9
230 CALL "DEFCHAR",9,64,96,160,160,96,96,160,144
240 CALL "DEFCHAR",10,10,12,10,9,9,250,252,248
250 CALL "DEFCHAR",11,80,48,80,144,144,80,62,30

```

```

260 CALL "DEFCHAR",12,255,255,255,255,126,126,126,255
270 CALL "DEFCHAR",13,255,254,252,0,0,0,0,0
280 CALL "DEFCHAR",14,255,127,63,0,0,0,0,0
290 CALL "DEFCHAR",15,255,255,255,0,0,0,0,0
300 CALL "DEFCHAR",16,0,0,31,6,1,0,0,0
310 CALL "DEFCHAR",17,0,0,255,0,255,96,96,240
320 CALL "DEFCHAR",18,0,0,255,6,248,97,97,243
330 CALL "DEFCHAR",19,255,255,255,255,255,255,255,255
340 CALL "DEFCHAR",20,85,170,170,255,255,255,255,255
350 SH#=CHR$(1)+CHR$(2)+CHR$(2)+CHR$(2)+CHR$(3)
360 HE#=CHR$(4)+CHR$(5)
370 OT#=CHR$(6)+CHR$(7)
380 OM#=CHR$(8)+CHR$(9)
390 OH#=CHR$(16)+CHR$(17)+CHR$(18)+CHR$(10)+CHR$(11)
400 OB#=CHR$(14)+CHR$(12)+CHR$(15)+CHR$(12)+CHR$(13)
410 LE#=CHR$(19)+CHR$(32)+CHR$(19)
420 BS#=CHR$(32)+CHR$(48)+CHR$(32)+CHR$(32)+CHR$(32)
430 SE#=CHR$(20)
440 BL#=CHR$(19)
450 ME#=STR$(ME)
460 HS#=STR$(HS)
470 SC#=STR$(SC)
480 REM          DRAW
490 CALL "COLOUR",0,20
500 CALL "COLOUR",1,180

```

```

510 PLOT 0,59,"SCORE"
520 PLOT 27,59,"LIVES"
530 PLOT 48,59,"HI SCORE"
540 IF FX=1 THEN GOTO 570
550 FX=1
560 PLOT 11,59,BB#
570 PLOT 38,59,ME#
580 PLOT 64,59,HS#
590 CALL "FILL",0,180,320,200,2
600 A=200: X=48
610 X1=60: Y1=61: X2=60: Y2=61
620 FOR X=1 TO 30 STEP 8
630 CALL "STPLOT",A,X,VARADR(LE#),1
640 NEXT
650 CALL "STPLOT",A-B,X,VARADR(OB#),3
660 CALL "STPLOT",A-B,X+B,VARADR(OH#),3
670 CALL "STPLOT",A+16,X+16,VARADR(OM#),3
680 CALL "STPLOT",A+16,X+24,VARADR(OT#),3
690 FOR X=0 TO 320 STEP 8
700 CALL "STPLOT",X,0,VARADR(SE#),2
710 NEXT
720 CALL "STPLOT",0,B,VARADR(SH#),3
730 CALL "STPLOT",X1,Y1,VARADR(HE#),3
740 REM MOVE
750 CALL &6000
760 IF INT(RND(1)*15)=3 THEN GOSUB 980
770 P=PEEK(&5FFF)
780 IF P<0 THEN P=P+256
790 IF P=0 THEN Y1=Y1-2
800 IF P=80 THEN Y1=Y1+2
810 IF P=80 THEN X1=X1-3
820 IF P=81 THEN X1=X1+3
830 X1=X1+LW
840 X1=X1-RW
850 IF X1<202 AND X1>190 AND Y1=71 THEN FB=1: GOTO 2400
860 IF X1>300 THEN X1=0
870 IF X1<0 THEN X1=300
880 IF Y1<8 THEN GOTO 1630
890 IF X1<18 AND X1>6 AND Y1=11 THEN FB=2: GOTO 2400
900 IF Y1<70 AND X1<222 AND X1>184 THEN 1240
910 IF Y1>170 THEN 1760
920 CALL "STPLOT",X2,Y2,VARADR(HE#),0
930 CALL "STPLOT",X1,Y1,VARADR(HE#),3
940 CALL "STPLOT",216,81,VARADR(OT#),3
950 CALL "STPLOT",216,73,VARADR(OM#),3
960 X2=X1: Y2=Y1
970 GOTO 750
980 REM WIND
990 PUT 12
1000 W=INT(RND(1)*3)
1010 S=INT(RND(1)*5)
1020 IF S=0 THEN WIS="Weak"
1030 IF S=1 THEN WIS="Mild"
1040 IF S=2 THEN WIS="Moderate"
1050 IF S=3 THEN WIS="Strong"
1060 IF S=4 THEN WIS="Gale force"
1070 IF W=0 THEN ?WIS;" wind from the Right": RW=S+1: SK=LW=0
1080 IF W=1 THEN ?WIS;" wind from the Left": RW=0: LW=S+1: SK=
1090 IF W=2 THEN ?"Wind dropped": LW=0: RW=0
1100 RETURN
1110 REM PICK UP
1120 CALL "STPLOT",X2,Y2,VARADR(HE#),0
1130 CALL "STPLOT",X1,Y1,VARADR(HE#),3
1140 PUT 12
1150 SU=INT(RND(1)*8)+10
1160 ?"You have successfully landed"
1170 ?"and picked up ";SU" survivors"
1180 FL=1
1190 FORT=1 TO 2000: NEXT T
1200 GOSUB 980
1210 P=&0B: Q=&0B
1220 CALL "STPLOT",X1,Y1,VARADR(HE#),0
1230 GOTO 740
1240 REM CRASH 1
1250 PUT 12
1260 ?"You have crashed into the Oilrig"
1270 IF FL=1 THEN ?"killing all of your survivors"
1280 ME=ME-1
1290 FORT=1 TO 2000: NEXT T
1300 FL=0
1310 PUT 12
1320 IF ME=0 THEN 1910
1330 ME#STR$(ME)
1340 PLOT 38,59,ME#
1350 X1=60: Y1=61
1360 P=&0B: Q=&0B
1370 CALL "STPLOT",X2,Y2,VARADR(HE#),0
1380 GOTO 480
1390 REM LAND WITHOUT
1400 CALL "STPLOT",X2,Y2,VARADR(HE#),0
1410 CALL "STPLOT",X1,Y1,VARADR(HE#),3
1420 PUT 12
1430 ?"Why land without any survivors?"
1440 FORT=1 TO 1000: NEXT T
1450 P=&0B: Q=&0B
1460 CALL "STPLOT",X1,Y1,VARADR(HE#),0
1470 RETURN
1480 REM LAND WITH
1490 PUT 12
1500 CALL "STPLOT",X2,Y2,VARADR(HE#),0
1510 CALL "STPLOT",X1,Y1,VARADR(HE#),3
1520 ?"Well done you have scored ";SU*10
1530 SC=SC+(SU*10)
1540 ST=ST+SU
1550 SC#STR$(SC)
1560 PLOT 11,59,SC#
1570 FL=0
1580 FORT=1 TO 2000: NEXT T
1590 P=&0B: Q=&0B

1600 CALL "STPLOT",X1,Y1,VARADR(HE#),0
1610 A#="A": B#="A"
1620 GOTO 750
1630 REM SEA
1640 PUT 12
1650 ?"You have crashed into the sea"
1660 IF FL=1 THEN ?"killing all of your survivors"
1670 ME=ME-1
1680 FL=0
1690 X1=60: Y1=61
1700 FORT=1 TO 1000: NEXT T
1710 PUT 12
1720 IF ME=0 THEN 1910
1730 ME#STR$(ME)
1740 PLOT 38,59,ME#
1750 GOTO 740
1760 REM TOO HIGH
1770 PUT 12
1780 ?"You have flown too high"
1790 IF FL=1 THEN ?"killing all of your survivors"
1800 ME=ME-1
1810 X1=60: Y1=61
1820 FL=0
1830 FORT=1 TO 2000: NEXT T
1840 IF ME=0 THEN 1910
1850 ME#STR$(ME)
1860 PLOT 38,59,ME#
1870 CALL "STPLOT",X2,Y2,VARADR(HE#),0
1880 PUT 12
1890 P=&0B: Q=&0B
1900 GOTO 960
1910 REM END
1920 CALL "STPLOT",X2,Y2,VARADR(HE#),0
1930 PUT 12
1940 ?"You have sent the rest of"
1950 ?"the crew to a watery grave"
1960 ?"You saved ";ST;"men."
1970 IF SC>HS THEN HS=SC
1980 HS#STR$(HS)
1990 PLOT 64,59,HS#
2000 SC=0
2010 ST=0
2020 SU=0
2030 FX=0
2040 A#=""
2050 B#=""
2060 ME=3
2070 ME#STR$(ME)
2080 SC#STR$(SC)
2090 FL=0
2100 FORT=1 TO 3000: NEXT T
2110 PUT 12
2120 ?"Do you want another go? (Y/N)"
2130 C#GET$(1)
2140 IF C#="Y" OR C#="y" THEN GOSUB 2540: GOTO 480
2150 IF C#="N" OR C#="n" THEN CALL "CLEAR": TEXT: END
2160 GOTO 2130
2170 REM INSTRUCTIONS
2180 TEXT
2190 PUT 31
2200 ?" North Sea Oil Rig Rescue"
2210 ?"
2220 ?"?" Steve Crick, May/83"
2230 ?"
2240 ?"?" Instructions"
2250 ?"
2260 ?"?" Due to a fault in the Oil Rig's legs"
2270 ?"it is in danger of collapsing. You are"
2280 ?"the pilot of the helicopter that is"
2290 ?"trying to save the workers. To make"
2300 ?"things worse there are unpredictable"
2310 ?"winds in the north sea, making your"
2320 ?"landings difficult."
2330 ?"?" You use the ARROW keys to move"
2340 ?" Any other key to stop"
2350 ?"?" Any key to play"
2360 A=GET$(1)
2370 GOSUB 2540
2380 PUT 12
2390 RETURN
2400 REM CUT DOWN
2410 IF FL=0 AND FB=1 THEN FB=0: GOTO 1110
2420 IF FL=1 AND FB=1 THEN FB=0: GOTO 1240
2430 IF FL=0 AND FB=2 THEN FB=0: GOSUB 1390
2440 IF FL=1 AND FB=2 THEN FB=0: GOTO 1480
2450 GOTO 750
2460 REM MACHINE CODE
2470 POKE &6000,&F7
2480 POKE &6001,&02
2490 POKE &6002,&32
2500 POKE &6003,&FF
2510 POKE &6004,&5F
2520 POKE &6005,&C9
2530 RETURN
2540 REM SKILL
2550 CALL "RESOLUTION",0,2
2560 TEXT
2570 PUT 31
2580 ?" Input Your Skill Level: "
2590 ?"
2600 ?"?" 0=Easy"
2610 ?"?" 1=Hard"
2620 ?"?" 2=Very Hard"
2630 ?"INPUT Level=",SK
2640 IF SK<0 OR SK>2 THEN 2570
2650 ?"?" Any key to play"
2660 C#GET$(1)
2670 PUT 12: GRAPH
2680 RETURN

```


NEWBRAIN NERVE CENTRE

by David Watt



General-purpose graph

THE FIRST program is a general-purpose line graph program developed by Robert Lewsley, designed to take the tedium out of drawing graphs from measurements he takes in the course of his work. Three options are available. You may input from keyboard or tape, or input from keyboard while simultaneously copying the data to tape. The program allows you to specify the titles, ranges and scale positions for both the x- and y-axes. Then you may input the x and y co-ordinates.

The program plots the graph as each set of co-ordinates is input, checking that the x value is greater than at the previous point. You can make multiple plots by specifying x and y values of zero to move the current position back to the start. Lines 2700 to 3160 display full instructions for using the program.

General-purpose graph.

```

1000 REM *****
1010 REM - general purpose graph program
1020 REM - copyright R. Lewsley 1983
1030 REM
1040 REM *****
1050
1060 ON BREAK GOTO 3210
1070
1080 REM *****
1090
1100 e$="Invalid input - try again"
1110 e2$="Text too long - limit = 20 characters."
1120 CLOSE#2:OPEN#0,4,"200":GOSUB 2700
1130 CLOSE#2
1140 PUT 31
1150 PRINT "Enter processing option (KB:
1150 KS:TI): ";:INPUT (" ");po$
1160 IF po$="KB" OR po$="kb" OR po$="KS"
1160 OR po$="ks" OR po$="TI" OR po$="ti"
1160 THEN 1180
1170 PRINT "Invalid processing option -
1170 try again":PUT 12:GOTO 1150
1180 PUT 31
1190 IF po$="KB" OR po$="kb" THEN po=1:G
1190 OTO 1330
1200 IF po$="KS" OR po$="ks" THEN po=2:
1200 GOTO 1280
1210
1220 REM - tape input
1230
1240 po=3:PRINT "Load input tape at corr
1240 ect point.":PRINT "Press play then
1240 press newline when ready":INPUT x$
1240 :OPENIN#2,1,"graph.data":GOTO 1330
1250
1260 REM - keyboard input plus save
1270
1280 PRINT "Load new tape at start point
1280 , and press":PRINT "record/play.":P
1280 RINT "Then press newline when ready
1280 .":INPUT x$:OPENOUT#2,1,"graph.dat
1280 a":GOTO 1330
1290

```

```

1300 REM - open graphics screen
1310 REM - and draw skeleton
1320
1330 PUT 31:OPEN#129,1,"w200":plotrng(1
1330 0,10),pla(0,0),mve(0,9,9)
1340 plotdeg,tby(-90),mby(9,9),tby(-90)
1350 plotmby(9,9),tby(-90),mby(9,9)
1360
1370 IF po=3 THEN LINPUT#2,x$:GOTO 1490
1380
1390
1400 REM - begin getting titles
1410 REM - and ranges
1420
1430 PRINT "Please enter title for graph
1430 "
1440 INPUT (" "); x$
1450 IF LEN(x$)<21 THEN 1490
1460 PUT 31:PRINT e2$
1470 GOTO 1440
1480
1490 plotrng(100,100)
1500 IF LEN(x$)=0 THEN x$=" "
1510 IF po=2 THEN PRINT#2,x$
1520 x=(100-LEN(x$)*3)/2-3:IF x<0 THEN x
1520 =1
1530 plotpla(x,94),x$
1540 plotpla(0,0)
1550 PUT 31
1560 IF po=3 THEN INPUT#2,x1,xh,y1,yh:G
1560 TO 1750
1570
1580
1590 ON ERROR GOTO 1630
1600 PRINT "Enter low and high values fo
1600 r X (nn,nn)"
1610 INPUT x1,xh
1620 GOTO 1640
1630 ON ERROR GOTO 0:PRINT e$:PUT 12:RES
1630 UME 1590
1640 IF xh-x1 > 0 THEN 1660
1650 PRINT e$:PUT 12:GOTO 1590
1660 PUT 31
1670 ON ERROR GOTO 1710
1680 PRINT "Enter low and high values fo
1680 r Y (nn,nn)"
1690 INPUT y1,yh
1700 GOTO 1720
1710 ON ERROR GOTO 0:PRINT e$:PUT 12:RES
1710 UME 1670
1720 IF yh-y1 > 0 THEN 1740
1730 PRINT e$:PUT 12:GOTO 1670
1740 ON ERROR GOTO 0
1750 PUT 31
1760 IF po=2 THEN PRINT#2,x1,xh,y1,yh
1770 IF po=3 THEN INPUT#2,x$:GOTO 1840
1780 PRINT "Enter title for X axis"
1790 INPUT x$
1800 IF LEN(x$)<21 THEN 1830
1810 PRINT e2$
1820 PUT 12:GOTO 1780
1830 IF LEN(x$)=0 THEN x$="X axis"
1840 x=(100-LEN(x$)*3)/2-3:IF x<0 THEN x
1840 =1
1850 plotpla(x,0),x$
1860 PUT 31
1870 IF po=2 THEN PRINT#2,x$
1880 IF po=3 THEN INPUT#2,x$:GOTO 1940
1890 PRINT "Enter title for Y axis"
1900 INPUT x$
1910 IF LEN(x$)<21 THEN 1940
1920 PRINT e$
1930 PUT 12:GOTO 1890
1940 IF LEN(x$)=0 THEN x$="Y axis"
1950 IF po=2 THEN PRINT#2,x$
1960 x=(100-LEN(x$)*3)/2-3:IF x<0 THEN x
1960 =1
1970 x=100-x
1980 FOR i=1 TO LEN(x$)
1990 plotpla(i,x),MID$(x$,i,1)
2000 x=x-4
2010 NEXT i
2020 plotcen(5,7),pla(0,0)
2030 x$=STR$(x1):x=LEN(x$)-1
2040 x$=LEFT$(x$,x)
2050 x=LEN(x$)-1
2060 x$=RIGHT$(x$,x)
2070 plotpla(-3,-6),x$
2080 x$=STR$(xh):x=LEN(x$)-1
2090 x$=LEFT$(x$,x)
2100 x=LEN(x$)-1
2110 x$=RIGHT$(x$,x)
2120 x=95-(x+6)
2130 plotpla(x,-6),x$
2140 x$=STR$(y1):x=LEN(x$)-1
2150 x$=LEFT$(x$,x)
2160 x=LEN(x$)-1
2170 x$=RIGHT$(x$,x)
2180 plotpla(-3,87),x$
2190 x$=STR$(yh):x=LEN(x$)-1
2200 x$=LEFT$(x$,x)
2210 x=LEN(x$)-1
2220 x$=RIGHT$(x$,x)
2230 plotpla(-3,-2),x$
2240 x=xh-x1:y=yh-y1
2250 plotrng(x,y),cen(0,0)
2260 xp=0-x/10:yp=0-y/10
2270 PUT 31
2280 FOR i=1 TO 10
2290 xp=xp+x/10:plotpla(xp,0),mve(xp,y/1
2290 00)
2300 yp=yp+y/10:plotpla(0,yp),mve(x/100,
2300 yp)
2310 NEXT i
2320 plotpla(0,0):x=x-1
2330
2340 PUT 31
2350 IF po<3 THEN 2460
2360
2370 REM - draw graph from tape data
2380
2390 INPUT#2,x$
2400 IF ASC(x$)=4 THEN PUT 31:PRINT "Dis
2400 play complete - press newline":PRIN
2400 T "to terminate run.":INPUT x$:STD
2400 P
2410 x=VAL(x$):INPUT#2,x$:yp=VAL(x$):G
2410 TO 2510
2420
2430 REM - begin drawing using keyboard
2440 REM - input
2450
2460 ON ERROR GOTO 2500
2470 PRINT "Enter X and Y coordinates (n
2470 n,nn)"
2480 INPUT xp,yp
2490 GOTO 2510
2500 ON ERROR GOTO 0:PRINT e$:PUT 12,12:
2500 RESUME 2460
2510 ON ERROR GOTO 0
2520 IF xp<0 OR yp<0 THEN 2570
2530 plotpla(0,0):x=x-1:IF po=1 THEN P
2530 UT 12:GOTO 2460
2540 IF po=3 THEN 2590
2550 x$=STR$(xp):PRINT#2,x$:x$=STR$(yp):
2550 PRINT#2,x$
2560 GOTO 2540
2570 IF po=3 THEN 2630
2580 IF xp=x1 AND yp=yh THEN 2600
2590 PRINT e$:PUT 12:GOTO 2460
2600 IF yp<1 OR yp>yh THEN 2590
2610 IF xp=xo THEN 2590
2620 IF po=2 THEN x$=STR$(xp):PRINT#2,x$
2620 : x$=STR$(yp):PRINT#2,x$
2630 x=xp
2640 xp=xp-x1:yp=yp-y1
2650 plotmve(xp,yp)
2660 GOTO 2540
2670
2680 REM - user instruction routine
2690
2700 PUT 31
2710 x$="General Purpose Graph Program"
2720 PRINT TAB(20-LEN(x$)/2);x$
2730 x$="(c) Copyright R. Lewsley 1983"
2740 PRINT TAB(20-LEN(x$)/2);x$
2750 PUT 10
2760 PRINT "Do you require instructions
2760 y/n ":INPUT x$
2770 IF x$="n" OR x$="N" THEN 3170
2780 IF x$="y" OR x$="Y" THEN PUT 31:GOT
2780 O 2800
2790 PUT 12:GOTO 2710
2800 PRINT "This program will draw a gra
2800 ph using"
2810 PRINT "either keyboard or tape input
2810 data."
2820 PUT 10:PRINT "If keyboard input is
2820 selected "
2830 PRINT "the program will initially r
2830 equest a"
2840 PRINT "title for the graph, then ask
2840 for the"
2850 PRINT "low and high range values of
2850 the"
2860 PRINT "X (horizontal) and Y (vertic
2860 al) axes."
2870 PUT 10
2880 PRINT "It will then request headin
2880 g s and scale"
2890 PRINT "mark values for these two ax
2890 es."
2900 PUT 10
2910 PRINT "At this stage a skeleton lay
2910 out"
2920 PRINT "will be drawn and you will b
2920 e asked"
2930 PRINT "to begin entering the X and
2930 Y values.":PUT 10,10,10
2940 INPUT ("press newline to continue"
2940 ):x$:PUT 31
2950 PRINT "Plotting will be done immed
2950 ately and"
2960 PRINT "each new set of X,Y values w
2960 ill be"
2970 PRINT "checked to ensure that X is
2970 greater"
2980 PRINT "than its previous value."
2990 PUT 10

```

(continued on page 158)

Bounce.

```
10 FOR a=1 TO 255:CLOSE#1:NEXT a
20 OPEN#1,0,1,"150":OPEN#2,0,2,"150"
30 OPEN#3,11,1,"#1,220":OPEN#4,11,2,"#2,
  220"
40 plot#4,rng(30,20),deg,pla(15,1),arc(1
  5,360),pla(15,2),fil
50 plot#3,rng(30,20),deg,pla(15,5),arc(1
  5,360),pla(15,7),fil
60 PRINT#1
70 FOR a=1 TO 200:NEXT a
80 PRINT#2
90 FOR a=1 TO 200:NEXT a
100 ON BREAK GOTO 120
110 GOTO 60
120 CLOSE#3:CLOSE#4
130 PUT 31,23,65
140 LIST
150 END
```

Rotate.

```
10 FOR a=1 TO 255:CLOSE#1:NEXT a
20 OPEN#0,0,"150":OPEN#1,0,1,"150":OPEN#
  2,0,2,"150"
```

```
30 OPEN#3,11,"220":OPEN#4,11,1,"#1,220":
  OPEN#5,11,2,"#2,220"
40 FOR c=3 TO 5
50 plot#c,rng(30,20),deg,pla(15,10)
60 NEXT c
70 plot#4,trn(120):plot#5,trn(240)
80 PRINT
90 FOR a=1 TO 4
100 FOR b=1 TO 5
110 FOR c=3 TO 5
120 plot#c,arc(b,90),tby(90),arc(b,90),t
  by(90)
130 NEXT c
140 NEXT b
150 FOR c=3 TO 5
160 plot#c,tby(90)
170 NEXT c
180 NEXT a
190 PRINT#1
200 FOR a=1 TO 50:NEXT a
210 PRINT#2
220 FOR a=1 TO 50:NEXT a
230 PRINT
240 FOR a=1 TO 50:NEXT a
250 GOTO 190
```

Format. 40.

```
2000 REM "format.40" by R. Lewsley.
2010
2020 REM Program to print hard copy
2030 REM lists of programs in a 40 col
2040 REM format for publication.
2050
2060 REM Input to this program is a tape
2070 REM "list" of the program created
2080 REM using the LIST command
2090 REM e.g.
2100 REM to create the tape load the
2110 REM program to be formatted
2120 REM then enter the following
2130
2140 REM openout#1,1
2150 REM list#1
2160 REM print#1,chr$(4)
2170 REM close#1
2180 REM the list tape is now ready for
```

```
2190 REM use as input to this program
2200
2210
2220 OPEN#0,0,"124"
2230
2240 CLOSE#0:OPEN#8,8,"1200"
2250
2260 PUT 31:PRINT TAB(35);"Format.40":TA
  B(65);"by R. Lewsley"
2270 PUT 10:PRINT TAB(20);"Load "list"
  tape in tape 1 and press play"
2280
2290 CLOSE#1:OPEN#1,1
2300 lc=99:pc=0:xs=""
2310 PUT#8,30,27,66
2320 LINPUT#1,a$
2330 IF a$=CHR$(4) THEN CLOSE#1:PUT 31:P
  RINT "READY":END
2340 IF LEN(a$)=0 THEN 2320
2350 m=40
2360 xs="" :CLEAR xs
2370 IF lc < 51 THEN 2430
```

```
2380 PUT#8,12
2390 pc=pc+1
2400 PRINT#8,"Page";pc:PUT#8,10
2410 lc=1
2420
2430 l=LEN(a$)
2440 IF l > m THEN 2510
2450 xs=xs+a$
2460 PRINT#8,x$
2470 lc=lc+1
2480 GOTO 2320
2490
2500
2510 i=m
2520 xs=xs+LEFT$(a$,i)
2530 l=l-i
2540 a$=RIGHT$(a$,l)
2550 PRINT#8,x$
2560 lc=lc+1
2570 m=35
2580 xs=""
2590 GOTO 2440
```

(continued from page 156)

```
3000 PRINT "Multiple plots may be made b
  y entering"
3010 PRINT "zero for both X and Y, this
  will move"
3020 PRINT "the current plotting positio
  n back to"
3030 PRINT "the start point."
3040 PUT 10,10,10
3050 PRINT "To terminate the program pre
  ss the"
3060 PRINT "stop key followed by newline
  ."
3070 PUT 10,10:LINPUT ("press newline to
  proceed")x$
3080 PUT 31
3090 PRINT "Three processing options are
  available."
3100 PUT 10:PRINT "KB - meaning keyboard
  input for"
3110 PRINT " immediate display only."
3120 PUT 10:PRINT "KS - meaning keyboard
  input for display"
3130 PRINT " plus save to tape for l
  ater use."
3140 PUT 10:PRINT "TI - meaning display
  previously saved"
3150 PRINT " data from tape input."
3160 PUT 10,10:LINPUT ("press newline to
  proceed")x$
3170 PUT 31:RETURN
3180
3190 REM CLOSDOWN ROUTINE
3200
3210 ON BREAK GOTO 0:IF pc=2 THEN 3240
3220 CLOSE#2:CLOSE#129
3230 PUT 31:PRINT "READY":END
3240 PRINT#2,CHR$(4)
3250 GOTO 3220
```

Format 40

This program by Robert Lewsley prints program listings in the 40-column format preferred by this magazine. I certainly found it useful in preparing some of the listings. Lines 2140 to 2170 describe how to store a program on tape prior to printing it. I found it better to specify a file name when storing my programs using the commands:

OPEN OUT#1,1,"progname" in place of the command on line 2140

Once your program is stored, Format 40 just has to be loaded and run to list it out. Line 2310 outputs some special initialisation characters for the Oki Microline 82a printer. It may have to be changed for your own printer.

Multiple screens

One of the most powerful features of the Newbrain is its ability to open a number of streams for one device. In particular, you can set up multiple screen displays which can be switched between at will.

Edward Thomas from Clapton, London E5, sent in two programs which demonstrate this very well and also

illustrate some of the features of the Newbrain high-resolution graphics. The first program, Bounce, opens two display streams on ports 1 and 2, and two linked graphics streams. It then draws a circle on each graphics stream and fills them in. Finally, the program goes into a loop where each stream is displayed in turn with a time delay between each display. The resulting effect is of a bouncing ball.

The second program is slightly more complex, using three streams. When run, Rotate will draw a four-pointed petal which, when completed, will start spinning anti-clockwise. Graphics use rather a lot of memory so three screens is the limit for this type of display with the standard system, and fully animated cartoons are out of the question.

Many useful facilities can still be provided with text displays. For example, help information and option menus could be stored on separate streams from the main display, to be called up as required. It is also possible to plot to one stream while the other stream is being displayed. Provided it does not take too long to plot the changes between displays, it should be possible to perform limited animation. 



User-defined I/O routines

A LITTLE PIECE of detective work has enabled Larry Carasco of Dollis Hill to produce the program which will allow the Spectrum stream facilities to be utilised. The streams will be used with the Sinclair Microdrives and networking board.

Anybody who already has some form of I/O device attached to their Spectrum might be able to put this program to use immediately. A channel consists of five bytes of code which define which routines are being used for I/O. The format is:

- Address of output routine,
- Address of input routine,
- File name.

The file name consists of a single character — of which more later. The Spectrum has a series of 19 streams which indicate where the channels are situated, of which 16 are available to the user. A stream is simply a 16-bit pointer to a channel. The address of the first channel is in Chans, 23631.

A stream which has been opened contains a pointer, which is 0 if the stream is closed. The channel to which it points is at

$$\text{CHANS} + \text{pointer} - 1.$$

Thus if Chans is 23734 and the pointer is 6 then the channel is at address 23739.

The Spectrum manual states that Strms is at 23568, the address of the first stream pointer. However, the streams at 23568, 23570 and 23572 are not available to the user so the address of the first user stream, called Stream 0, is 23574. The address of any given stream is at

$$23574 + 2 \times \text{stream number}$$

where the stream is any number between 0 and 15. The contents of this address plus Chans minus 1 gives the channel its uses. Table 1 shows the stream addresses, their pointers and the channel data they point to.

Streams 0 to 3 are normally used by the Spectrum for display and input. Streams 0 and 1 point to the same channel and are used to write to screen lines 22 and 23 and also to read the keyboard. Stream 2 is used by the Print and List statements to output to lines 0 to 21. Stream 3 is used by the printer; LPrint and LList use this stream.

Examining the channels which these

streams point to clarifies the I/O structure. Table 2 shows the channels which the Spectrum sets up on default. The channel data is situated at Chans to Chans + 19.

The subroutine at 2548 is a conventional output routine, and 4264 is an input routine which returns characters entered from the keyboard. The routine at 3969 probably requires extra hardware to function as intended. The routine at 5572 causes error J to occur: you may not input from this device. The file name is a mnemonic for the channel's I/O port:

K = keyboard and lines 22,23
S = screen, lines 0 to 21
R = RS-232 (?)
P = Printer

No other file name is valid, and only K, S and P may be specified by the user.

Although three channels use the same output routine, Basic sets various flags to indicate which channel is being serviced. The file name decides which flags are set. This means, for instance, that when using Channel P the output is not echoed to the screen. Using this method of streams and channels reduces the amount of memory required to handle many different I/O routines.

All the streams using a particular channel have their pointers set to the same value, and all 16 streams could use the same channel — though it would be rather pointless. Five bytes of data would then service the entire I/O network. The address of the channel currently in use is held in Curchl, 23633.

A specific stream can be selected by inserting a # sign, CHR\$ 35, and a stream number. For example.

PRINT #3; "Hello!"

will output to the printer. Using table 2 to help establish what effect the different channels have, try out different streams for this example. Only streams 0 to 3 are valid at present.

Inkey \$,Print and Input may all use stream values. Now try

LPRINT #2; "Hi!"

A command which normally writes to the

printer has been told to use Stream 2, which in turn has directed it to Channel S.

All I/O statements, except Verify, Save and Merge, always use streams but because they use default values when one has not been specified this is, perhaps, not apparent. Print defaults to Stream 2, LPrint to Stream 3 and so on. When you specify a particular stream you are directing the I/O of that statement to a chosen channel.

Contrary to the insistence of the manual, the Open and Close statements can be used without extra hardware, as you may have already discovered. When you use them you are actually defining which channel you wish that stream to use. The format is:

OPEN #stream number, file name

The valid file names are K, S and P.

Try entering

PRINT #5; "Illegal"

You should try to get an Error 0 report, but if you first enter

PRINT #5, "S"

the text should appear on the screen. What you have done is opened Stream 5 and instructed it to use Channel S. Any stream number between 4 and 15 will work.

Streams 0 to 3 will also work, but you will be altering the normal system I/O configuration. You should always take care when altering them or you may lock yourself out of the system. If you want a demonstration,

OPEN #3, "S"

is fairly safe and will cause printer output to be rerouted to the screen.

The Close statement naturally enough closes down the chosen stream. For streams 4 to 15 it resets their pointers to 0, indicating that the stream is disconnected — see table 1. Closing streams 0 to 3 results in returning their pointers to their default

Channel	Output address	Input address	File name
0	2548	4264	K
1	2548	5572	S
2	3969	5572	R
3	2548	5572	P

Table 2: The Spectrum's four channels.

Stream	Address	Pointer value	Channel address
0	23574	0001	23734
1	23576	0001	23734
2	23578	0006	23739
3	23580	0016	23749
4	23582	0000	CLOSED
5	23584	0000	CLOSED
6	23586	0000	CLOSED
7	23588	0000	CLOSED
8	23590	0000	CLOSED
9	23592	0000	CLOSED
10	23594	0000	CLOSED
11	23596	0000	CLOSED
12	23598	0000	CLOSED
13	23600	0000	CLOSED
14	23602	0000	CLOSED
15	23604	0000	CLOSED

Table 1. Breakdown of stream data, values in decimal.

values. Be careful when closing streams down: an unfortunate program bug crashes the system when you attempt to Close a stream which was never Opened.

Even though you now know how the streams and channels operate constructing your own I/O routines is not as simple as it might be. At present you have only three channels you may use, K, S and P. Any other letter is discarded as an illegal file name.

To get round this you must create your own channel by a back-door method. The program will create just such a channel. Line 20 allocates five bytes of memory for the channel data by setting up a dummy line 0. The addresses of your new I/O routines are Poked into this dummy line. Finally you must give the channel a legal file name, otherwise the system will fail to recognise it as legal and might crash when you come to close down a stream.

To allow easy access to the screen the channel can be called S. When you open a stream to Channel S using the conventional method it will still think you mean the original S channel. With the channel thus set up we simply Poke in the stream's new pointer whose value is arrived at by:

address of new channel — CHANS+1.

To initialise any other stream to this channel just use:

LET cn = new stream number : GOTO 110
If any of the variables has been altered, run the program again specifying the new stream. The other streams will remain intact. To test the program try the following:

```
LET a =USR "a": POKE a,62: POKE
a+1,65:POKE a+2,195
POKE a+3,244:POKE a+4,9
```

Now run the program and answer the prompts as follows:

Stream number: 3

Output routine address: USR "a"

User-defined I/O routines.

```
10 REM 123456
20 POKE 23756,0: POKE 23760,14
30 INPUT "Open stream no? ";cn
40 INPUT "Output routine address
50 INPUT "Input routine address
60 LET ch=PEEK 23631+PEEK 2363
2*256
70 LET z=23761
80 LET a=z: LET x=out: GO SUB
500
90 LET a=a+2: LET x=in: GO SU
B 500
100 POKE a+2,CODE f#
110 LET a=23574+cn*2
120 LET x=1+z-ch: GO SUB 500
499 STOP
500 POKE a,x-INT (x/256)*256
510 POKE a+1,INT (x/256): RETUR
N
```

Input routine address: 5572

File name: "S"

Now try LList. Every character should appear as A: you have routed the printer stream through your own output channel which will only allow As to be printed. You could

POKE USR "a"+1

with any other ASCII character code to output that character instead. If you try

PRINT INKEY\$ #3

you will get Error J.

To erase the main program but keep initialised streams enter the following:

```
LET var = PEEK 23627 + PEEK 23628 x
256
```

```
LET a = var - 23771
```

```
POKE 23769,a - INT (a/256) x 256
```

```
POKE 23770,INT (a/256)
```

Delete line 20 and the program is deleted, save for line 0 which cannot be deleted because it contains the channel data.

This program only sets up a new channel for a stream to use. You must define your own I/O routines to service the channel.

These routines could be used for just about

any peripheral you can attach to a Spectrum, be it a networking system, a Teletype or even another Spectrum. All you need to know is that the alternative register set should not be used, the output routine should output the value in the A register — preserving it, if possible — and the input routine should return with the Carry flag set if a valid character has been received, reset if not. Unless the interrupt is disabled by your routine the routine is interruptable, so try to avoid any unorthodox stack handling.

Smooth scroll

A VARIABLE degree of scroll is provided by this routine for the Spectrum, written by Paul Maycock of Bristol. The program loads the machine-code routine in the Data statement and then saves it for future use as a subroutine in any Basic program. The routine itself when called will scroll, so to scroll one line would require a For-Next of 8.

(continued on page 163)

Smooth scroll.

```
100 REM by P Maycock 18/1/83
110 DATA 33,0,65,17,0,64,6,3,19
7,6,8,197,6,7,197,1,32,0,237,176
,1,224,0,9,229,213,225,9,229,209
,225,193,16,236,1,224,7,237,66,1
,32,0,237,176,1,32,0,237,66,229
120 DATA 209,1,0,1,9,193,16,209
,1,0,7,213,225,9,1,32,0,229,237,
66,229,209,225,1,32,0,237,176,1,
0,1,213,225,9,193,16,177,33,224,
87,1,0,32,113,35,16,252,201
130 INPUT "Start Address ?";s
140 FOR f=0 TO 97
150 READ a
160 POKE s+f,a
170 NEXT f
190 SAVE "scroll mc"CODE s,98
200 INPUT "Press ENTER to verif
y"; LINE a#
210 VERIFY "scroll mc"CODE
```

Bridge hand.

```
10 DIM A(52)
20 FOR I=1 TO 52
30 LET A(I) = INT ((I-1)/13)
40 NEXT I
50 FOR I = 52 TO 2 STEP -1
60 LET T=INT(RND*I+1)
70 LET R=A(I)
80 LET A(I)=A(T)
90 LET A(T)=R
100 NEXT I
110 FOR K = 0 TO 3
120 PRINT "NORTHEAST SOUTHWEST "
(K*5+1 TO K*5+5);
140 FOR J= 0 TO 3
150 PRINT
160 FOR I = 1 TO 13
170 IF A(J*13+I)<>K THEN GOTO 210
180 LET R$ = " " + "234567890JQKA"
(I)+"SHDC"(J+1)+" "
```

(listing continued on page 163)

(continued from page 161)

When the program is run it asks for the start address where the code is to be Poked. It can be anywhere in the free RAM but is most likely to be 32500 in a 16K Spectrum or 65200 in a 48K machine. To call the routine use Randomise User s, where s is the value which you input when the program asks for the start address.

Correlation coefficient

A PROGRAM for the 16K Spectrum by Michael Coombes of Caerleon, Gwent calculates the product-moment correlation coefficient of a set of data. It will be useful to anyone studying or using statistics. Instructions are included in the program.

Bridge hand

If you enjoy bridge, this short program by P A Smith will help you to keep your bidding up to the mark. It is written for the unexpanded ZX-81 and could easily be expanded to print a series of hands.

Side scroll

AN IMPROVEMENT to the routine by C D Henderson, published in the March issue, comes from M J V Moreton of Cambridge. He points out a number of faults in Mr Henderson's routine:

- Some of the scans in lines 16 to 24 are not scrolled.
- The screen attributes are not scrolled.
- Items which disappear from the screen reappear at the right-hand side in higher.

This routine does not attempt to wrap the screen around but it does avoid the faults of the earlier version. The routine may be called by

LET variable = USR (32556)
and may be relocated elsewhere in RAM.

Side scroll.

```
10 CLEAR 32555
20 LET sum=0
30 FOR n=32556 TO 32599
40 READ a: POKE n,a
50 LET sum=sum+a
60 NEXT n
70 IF sum=2944 THEN STOP
80 PRINT FLASH 1;"Error"
90 DATA 22,0,33,0,64
100 DATA 1,32,0,30,192
110 DATA 114,9,29,32,251
120 DATA 58,141,92,30,24
130 DATA 119,9,29,32,251
140 DATA 33,1,64,17,0
150 DATA 64,1,0,27,237
160 DATA 176,18,33,255,87
170 DATA 22,0,114,201
```

Bridge hand listing summary.

Line 10 — Initialises array A(52) to represent cards.	Line 140 — Scans four suits.
Lines 20 to 40 — With 13 each of 0-3 to represent players, in arbitrary order.	Line 160 — Scans 13 cards per suit.
Lines 50 to 100 — Form random permutation.	Line 170 — Checks if card belongs to hand.
Line 110 — Prints four hands.	Line 180 to 190 — Creates strings R\$ for card.
	Line 200 — Prints card.

(continued from page 161)

```
190 IF R$(2) = "0" THEN LET
  R$(1) = "1"
200 PRINT R$;
210 NEXT I
220 NEXT J
230 PRINT
240 NEXT K
```

Correlation coefficient.

```
10 REM Product Moment
  Correlation Coefficient
  M.Coombes 1983
12 PAPER 0: BORDER 0: CLS : IN
  K 7
15 LET sumxx=0: LET sumyy=0: L
  ET sumx=0: LET sumy=0: LET sumxy
  =0
20 INPUT "How many values of x
  ? ";nx
30 BEEP .1,1
35 DIM x(nx): DIM y(nx)
40 PRINT AT 15,0;"Please enter
  all the values of x,each fo
  llowed by ENTER..."
50 FOR f=1 TO nx
60 INPUT x(f)
65 BEEP .1,1
67 LET sumx=sumx+x(f)
68 LET sumxx=sumxx+(x(f)^2)
70 PRINT AT 19,0;"
  ";AT 19,0;"x
  value ";f;" = ";x(f)
80 NEXT f
90 CLS
```

```
100 BEEP .4,10
110 PRINT AT 15,0;"Please enter
  all the values of y,each fo
  llowed by ENTER..."
120 FOR f=1 TO nx
130 INPUT y(f)
140 BEEP .1,1
141 LET sumxy=sumxy+(x(f)*y(f))
142 LET sumyy=sumyy+(y(f)^2)
145 LET sumy=sumy+y(f)
150 PRINT AT 19,0;"
  ";AT 19,0;"y
  value ";f;" = ";y(f)
160 NEXT f
165 BEEP .4,10
170 LET suma=sumx^2: LET sumb=s
  umy^2
200 REM *Calculate Coefficient*
210 LET co=(sumxy-((sumx*sumy)/
  nx))/(SQR ((sumxx-(suma/nx))*(su
  myy-(sumb/nx))))
300 REM *Print Answer*
310 PRINT AT 9,0;"The product m
  oment correlation coefficient fo
  r your data is:"
320 PRINT INK 5;co
330 PRINT AT 15,0; INK 6;"Press
  A to enter new data""Press B
  to exit"
340 IF INKEY$="a" OR INKEY$="A"
  THEN RUN
350 IF INKEY$="b" OR INKEY$="B"
  THEN STOP
360 GO TO 340
```



SHARP MZ-80B

Basic listing

ONE SHORTCOMING of the Sharp MZ range of computers is that they are intended to be interfaced only with Sharp's own range of printers — and they are expensive. One of the advantages of the MZ-80B is that it is available with an IEEE-488 interface which conforms exactly to the IEEE spec, unlike those offered by many of Sharp's rivals. This feature makes the MZ-80B a reliable instrument controller.

In this role it is clearly an easy matter to attach any low-cost printer to the IEEE bus to provide a convenient hard-copy medium for test results, etc. It would be convenient to list Basic programs in the same way. Unfortunately, the List commands provided in Sharp's Basic dump either to the screen or on to the printer interface if it is present.

The program by Jack Hale of Manchester overcomes this limitation when appended to an existing Basic program. It will list the program via the IEEE bus on an Epson MX-82 printer. It may be modified to suit other printers by changing the control characters in the WRT statements.

Rem lines are detected and printed in double-width characters centred in the line to form titles. This facility may be removed if not required by changing line 61200 to

```
WRT 4, OP$
```

and omitting lines 61230 to 61300.

It is convenient to position the listing routine at the end of the Basic program, hence the high line numbers. Listing of this routine may be suppressed so that only the main program is listed changing line 61500 to

```
IF PEEK (K + 2) + PEEK (K + 3) * 256
< 60000 THEN J = K: F1 =
0: GOTO 60400
```

The routine steps through the Basic area of memory line by line. The contents of a line are built up into a string OP\$ which is sent to the printer when complete, together with the line number. Commonly

used Basic words are held in memory in token form as one or two ASCII characters. The tokens deciphered by stepping through a look-up table held in the interpreter. This task is performed using a machine-code subroutine which is loaded in lines 60020 to 60095. The equivalent Basic is unacceptably slow.

Listing is initiated by entering Run 60000. To list the entire program respond to the prompt with 0. Responding with a higher number will result in that and subsequent lines being listed. To terminate the listing before the end of the program has been reached, press Break.

High-resolution dump table 1.

Bits	Screen 1,0	Screen 1,1
11	Red	Orange
10	Blue	Magenta
01	Yellow	Cyan
00	Green	Buff

Basic listing.

```
60000 REM *** BASIC LISTING PROGRAM ***
60001 REM
60002 REM Jack Hale - UMIST - 1982
60003 REM
60004 REM
60005 REM
60010 LIMIT $FEFF
60020 FOR I=65280 TO 65359:POKE I,0:NEXT I
60030 POKE $FF10,38:POKE $FF11,0:POKE $FF12,255
60040 POKE $FF13,230:POKE $FF16,127
60045 POKE $FF18,61:POKE $FF19,0
60050 POKE $FF20,42:POKE $FF1D,2:POKE $FF1E,255
60055 POKE $FF21,75
60060 POKE $FF22,203:POKE $FF23,121
60065 POKE $FF28,202:POKE $FF29,32:POKE $FF2A,255
60070 POKE $FF30,184
60075 POKE $FF31,202:POKE $FF32,64:POKE $FF33,255
60080 POKE $FF38,4
60085 POKE $FF39,195:POKE $FF3A,32:POKE $FF3B,255
60090 POKE $FF40,34:POKE $FF41,8:POKE $FF42,255
60095 POKE $FF47,201
60110 LIMIT $FEFF
60130 CONSOLE C40
60140 PRINT " PROGRAM LISTING"
60160 WRT 4,CHR$(27),"A",CHR$(8),CHR$(18),CHR$(20)
60180 PRINT CHR$(1):INPUT " INPUT START LINE NO ";LS
60200 J=29276:F1=0:F3=-1
60300 K=PEEK(J)+PEEK(J+1)*256
60350 IF PEEK(J+2)+PEEK(J+3)*256<LS THEN J=K:GOTO 60300
60370 CONSOLE C80
60400 K=PEEK(J)+PEEK(J+1)*256:OP$=""
60500 FOR I=J+4 TO K-1
60600 IF PEEK(I)=34 THEN F3=F3*(-1)
60650 IF (F1=1)+(F3=1)+(PEEK(I)<128) THEN OP$=OP$+CHR$(PEEK(I)):GOTO 61080
60670 POKE $FF02,50:POKE $FF03,24
60700 IF PEEK(I)=128 THEN I=I+1:POKE $FF02,145:POKE $FF03,22
60800 POKE $FF00,PEEK(I)
60930 USR($FF10)
60930 KW$=""
60970 N=PEEK($FF08)+PEEK($FF09)*256
61000 IF PEEK(N)<128 THEN KW$=KW$+CHR$(PEEK(N)):N=N+1:GOTO 61000
61030 KW$=KW$+CHR$(PEEK(N)-128)
61050 IF KW$="REM" THEN F1=1
61070 OP$=OP$+KW$
61080 NEXT I
61100 WRT 4,PEEK(J+2)+PEEK(J+3)*256," ";
61200 IF LEFT$(OP$,3)<>"REM" THEN WRT 4,OP$:GOTO 61400
61230 IF LEN(OP$)=4 THEN WRT 4,"REM":CHR$(13):GOTO 61400
61270 FOR A=4 TO 80:IF MID$(OP$,A,1)=" " THEN NEXT A
61280 TL=LEN(OP$)-A
61290 IF TL*2>71 THEN WRT 4,OP$:GOTO 61400
61300 WRT 4,"REM":SPACES((71-TL*2)/2):CHR$(14):RIGHT$(OP$,TL+1);
61400 PRINT PEEK(J+2)+PEEK(J+3)*256," ";OP$
61500 IF PEEK(K)+PEEK(K+1)<0 THEN J=K:F1=0:GOTO 60400
61600 END
```

High-resolution dump.

```
10 PMODE 4:SCREEN 1,1
20 FOR A = 0 TO 31
30 PRINT #-2,CHR$(27);"3";CHR$(24);
40 PRINT #-2,CHR$(27);"K";CHR$(191);CHR$(0);
50 FOR B = 1 TO 191
60 P = PEEK (7680 - (B * 32 - A))
70 PRINT #-2,CHR$(P);:NEXTB
80 PRINT #-2,CHR$(10);:NEXTA
90 GOTO 90
```

DRAGON 32

High-resolution dump

THIS PROGRAM by S J Combes of Bishop's Stortford, Hertfordshire works for PMode 3 and 4. It executes in 2.5 minutes although it does not use machine code, and dumps the screen to an Epson MX-80 MkIII. This speed improvement over the program by R A Shackleford, published in the April issue of *Practical Computing* is achieved by Peeking high-resolution screen memory and sending the values direct to the printer.

In PMode 4 the screen is stored as 192 horizontal lines of 32 bytes. If a bit is set the corresponding pixel is also set. The printer expects the bytes to be aligned vertically, which means that the picture must be printed on its side. This is a

welcome advantage as it allows side-by-side printout of page 1 and page 2 graphics.

PMode 3 is more complicated. Two bits are used for each pixel and are coded as shown in table 1. Areas of red or orange appear black; areas of blue, magenta, yellow and cyan appear grey; and green/buff appears white. To invert the picture add the line:

65 P = 255 - P

To print both screens side by side add:
72 PRINT # -2, CHR\$(27); "K"; CHR\$(191); CHR\$(0);

74 FOR B = 1 TO 191

76 P = PEEK (13824 - (B * 32 - A))

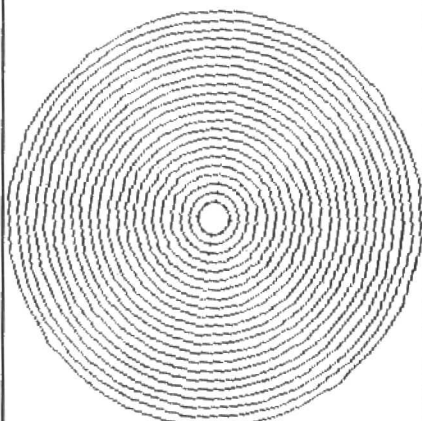
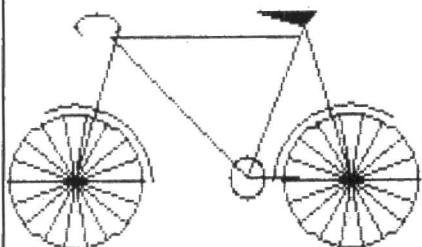
78 PRINT #2, CHR\$(P); :NEXT B

It is not necessary to merge the program to copy the screen since the high-resolution screen remains in memory after a New or a CLoad. Note that circles become ellipses when dumped. They must therefore be given a height to width ratio of 0.83 when drawn.

High resolution dump table 2.

Line	Function
10	Displays what is being printed on screen
30	Sets line spacing to 24/216 in.
40	Enter bit image mode for 191 characters
60	Peek screen location

High-resolution dump sample output.



NASCOM

THIS SHORT routine by G Winstanley of Stoke-on-Trent enables professional standard mainframe or minicomputer communication via the RS-232 serial interface. Input/output is achieved via the standard Nascom Uart, and connection to and from the Modem is to the user-available serial I/O socket.

Using the selection links of LSW2, it is possible to have speed selections of 110,300 and 1,200 baud. The only hardware modification required, and that is optional, is the connection of an acoustic warning device to bit 4 port 00H, the unused bit of the keyboard port. Control-R reinitialises the program.

The program has been kept short with the inclusion of some monitor subroutines and one restart instruction. The Blink routine maintains a blinking cursor and waits for input. It returns with the character in register A, and it is possible to detect whether input has occurred via serial in, or keyboard. One possible problem could arise if your host computer makes use of special control codes. Blink services certain control codes, such as Cursor Up, Down, etc. within itself.

Xout performs Uart output with handshaking. It is only necessary to place the output character in location OUTP prior to calling. The Kbd routine prints a character to the Nascom screen. The Cler routine clears the screen and RST 28H prints on the screen the ASCII string following, up to the first null character 00H.

Nascom as terminal.

```

0010 ; ** NASCOM COMPUTER TERMINAL PROG. **
0020 ;ROUTINES PERMIT TWO-WAY COMMUNICATION
0030 ;VIA RS232
0040 ;NASCOM BECOMES A PROFESSIONAL STANDARD
0050 ;REMOTE TERMINAL FOR FULL DUPLEX USE
0060 ;
0070 ;G.Winstanley OCTOBER 1982
0080 ;
4000 0100 ORG 4000H
0110 ;
4000 0078 0120 BLINK EQU 0078H ;CURSOR & INPUT SUB.
4000 0706 0130 XOUT EQU 0706H ;OUTPUT ROUTINE
4000 0030 0140 KBD EQU 0030H ;SCREEN O/P
4000 080A 0150 SCRE EQU 080AH ;TOP LINE POSITION
4000 08CA 0160 POS EQU 08CAH ;NEW CURSOR POS.
4000 03FA 0170 CLER EQU 03FAH ;CLEAR SCREEN
4000 0C29 0180 CURS EQU 0C29H ;CURSOR POS. LOCATION
4000 0C28 0190 OUTP EQU 0C28H ;OUTPUT BUFFER
0200 ;
4000 210640 0210 INIT LD HL,TINI ;SOFTWARE PIO RESET
4003 E5 0220 PUSH HL
4004 ED4D 0230 RETI
4006 CDF403 0240 TINI CALL CLER ;CLEAR SCREEN
4009 210A08 0250 LD HL,SCRE ;INIT.MESSAG
400C 22290C 0260 LD (CURS),HL
400F EF 0270 RST 28H ;PRINT @ TOP
4010 2A2A204E 0280 TABL DEFB '** Nascom Computer Terminal **'
6173636F
6D20436F
6D707574
65722054
65726D69
6E616C20
2A2A
402E 00 0290 DEFB 00H
402F 21CA08 0300 TASS LD HL,POS ;PLACE CURSOR
4032 22290C 0310 LD (CURS),HL
4035 CD7800 0320 TAPP CALL BLINK ;READY TO INPUT
4038 2010 0330 JR NZ DQT
403A FE12 0340 CP 12H ;RESET=CONTROL R
403C 28C2 0350 JR Z INIT ;RE-INITIALISE
403E FE07 0360 CP 07H ;BELL CODE?
4040 2812 0370 JR Z BELL
4042 32280C 0380 LD (OUTP),A ;FOR OUTPUT
4045 CD0607 0390 CALL XOUT ;NASCOM O/P
4048 18EB 0400 JR TAPP
404A 00 0410 DQT NOP
404B FE07 0420 CP 07H ;BELL CODE?
404D 2805 0430 JR Z BELL
404F CD3000 0440 CALL KBD ;OUTPUT-SCREEN
4052 18E1 0450 JR TAPP ;BACK TO COMMUNICATE
4054 3E10 0460 BELL LD A,10H ;BIT 4 PORT 00=BELL
4056 D300 0470 OUT (00H),A
4058 11FF1F 0480 LOPP LD DE,1FFFFH ;APPROX 0.1 SEC
405B 18 0490 DEP DEC DE
405C 7A 0500 LD A,D
405D B3 0510 OR E ;COUNT=ZERO ?
405E 20FB 0520 JR NZ DEP
4060 18D3 0530 JR TAPP

```

ALL THE IBM PC BOOKS received so far are American, not surprising since in America the machine has a larger share of the market than in the U.K. Systems stretch from 16K cassette-based models used as home computers, to 544K models with hard discs used in major corporations. The range of IBM PC books, not surprisingly, matches the range of PC applications.

At the beginners' level there are already a number of books which aim to teach simple Basic programming. Probably the best of these is David A Lien's *Learning IBM Basic for the Personal Computer*. This features the PC in its Charlie Chaplin disguise, and is illustrated with cartoons.

David Lien is the serious, highly respected author of the invaluable *The Basic Handbook*, published by Compusoft. However, his IBM book wallows in the worst excesses of the down-home American style. You do not so much read the book as sit grimacing while it talks off the page at you. Awwk!, Shew!, Wow! and Oooops! are typical of its interjections. Many sentences are somewhat lacking from a grammatical point of view and not all the jokey analogies strike home.

However, the book is sound from the computing point of view. It contains lots of short example programs and all the ones I tried actually worked. The book is, therefore, probably a good choice if you can stand the style.

Hands-On Basic for the IBM Personal Computer by Herbert Peckham is a machine-specific version of a previous book, *Basic: A Hands-on Method*, with graphics and sound sections added. The style is much more serious than in Lien's book. It is textbooky, perhaps because Peckham used to be a professor at Gavilan College. Each chapter has about seven sections: objectives, discovery exercises, discussion, program examples, problems, practice test. It could be used in a classroom or for self study, but it is not as informative, as readable, or as easy to dip into as Lien's volume.

Basic for Business for the IBM Personal Computer is also organised like a textbook, complete with ruled blank pages for you to write answers to set exercises. It is not a particularly entertaining book, but Alan Parker has managed to write in a straightforward way with as little jargon as possible. It should therefore be accessible to the average small businessman. The main topics covered are calculation, data entry and file keeping, though there is also a chapter on using VisiCalc.

The illustrations include a lot of flowcharts and sample runs of programs. Many of the programs included are quite long but excessively well documented, and do fairly useful things like sorting or writing receipts. Naturally all the examples and analogies are businesslike. While the thought of businessmen running their companies on home-written Basic software fills me with horror, at least working through the book would give them some

IBM PC books

Jack Schofield makes his selection.



idea of how to evaluate packaged software.

IBM Data Files: A Basic Tutorial is like *Basic for Business*, only more so. The question-and-answer bits are no more than quick quizzes — no bad thing — and the text is even easier to follow. It contains a lot of sensible hints and tips that obviously come from experience because they only occur to people who have tried to explain computing to half-wits.

On the other hand, as well as illustrative examples the book also contains some very long useful programs. A Home Inventory System, for example, comprises 18 pages of listings and the author shows how it could be converted into a back-order system for small business use. The programs are modular, logical and well documented; they look as though they should work. So although *IBM Data Files* sounds more limited than other works, it is as educational and probably more useful than its rivals.

IBM Basic for Business and Home starts right at rock bottom with, "What is a Computer?, What is ROM?" and similar questions. The answers are very short so the book moves at a fast pace. The main part of the book is a guide to Basic keywords, which provides shorter and simpler accounts than are found in IBM's own

manuals. The last part of the book deals with practical programming and then there are some very useful appendices — lists of commands and such like.

What the total package provides is a sort of potted version of the manuals, so the beginner can actually start computing more or less straight away. The book is unlike others reviewed earlier in that the author assumes the use of a proper PC set-up, including disc drives and a printer. He appreciates that most of the time users will be running packaged software. It is a sensible and useful book, which IBM ought to pack with its machines; it would certainly save their dealers more than its cost in time.

For people who just want Basic programs to type in there are two volumes on offer. *Some Common Basic Programs for the IBM Personal Computer* is the familiar Osborne/McGraw-Hill book. It is available in other editions for other micros including Pet, Atari, TRS-80 and the Apple II. There are 76 programs in all, which fall into four main categories: finance, maths, statistics and utilities. Examples include the usual interest-rate calculations, angle conversion, binomial distribution and sorting.

However, having reviewed the Atari version of this book in *Practical Computing* a few months ago I have two

(continued on next page)

(continued from previous page)

comments to make. The conversion of the programs to the specific machines shows the minimum of effort. Things like function keys, error trapping, graphics and sound are ignored. Also, before you buy the book think about whether you really need the programs.

The second book of programs, *Useful Basic Programs for the IBM PC*, is about half as big. It contains 65 programs organised into seven chapters and three appendices, and covers the usual topics such as maths, home finance and data analysis. Most of the programs are only about 20 lines, half a page, long and again, unless you are an absolute beginner, you could probably write them just as easily yourself.

There are a number of books which deal with operating the IBM PC in general without being tied to Basic, though of course Basic programming is a major feature of most of them. *IBM Personal Computer: An Introduction to Programming and Applications* is aimed at novices, and about 80 percent of the content is about Basic. In most of the book, however, the focus of attention is not on the language itself but on applications, such as, filing, graphics, word processing, games and science.

One version of the book comes packed in a box with a disc containing all the programs. As you might expect in a beginners' book, however, the level is pretty trivial. Overall the book seems adequate, though it is hard to enthuse over it. Perhaps I was just put off by the *Popular Computing* review quoted in large red letters across the cover: "... you should definitely buy [this] book ...". It would not be my first choice.

Using Your IBM Personal Computer is Lon Poole's effort. He has been involved in the production of books for other machines including the *Apple II User's Guide* and *Your Atari Computer*, which is much the best Atari book available. His IBM PC effort is similar in approach, and also very good. If you had no other documentation at all you could probably learn the PC from this book. It deals with setting up the system, discs and disc copying and even batch processing in part one, since this is where the average PC user will start. It is only in part two it moves on to Basic programming.

Actually Poole takes some stuff for granted, but he is excellent on the things that are not intuitively obvious, such as numeric strings and formatting output, and things that are particular to the PC. He is also very good on sound and graphics and working the printer. The book has some useful appendices including a Basic summary, tables of screen characters and codes, and an unusually good index. It is a very useful book for someone who is new to the IBM PC, but not necessarily new to computing. It would be ideal for easing the transition from, say, a Vic-20.

T G Lewis's book *Using the IBM*

Personal Computer is even less devoted to Basic. In fact, the Basic interpreter gets less space than using VisiCalc, and only slightly more than the Pascal compiler. Unlike Poole, however, Lewis does not assume a familiarity with computing. The first chapter is "What can computers do?" He deals with the subject briefly but intelligently. All through the book Lewis manages to produce the best kind of technical writing: he is specific without being verbose, readable without being patronising.

Of course he is not without idiosyncrasies. No-one christened Theodore Gyle, who dedicates a computer book "To life in the Oregon hills", can be completely normal. However, he is writing for people with disc-based systems who want to do serious things and run packaged software, and he never loses sight of this.

However, the discussions of VisiCalc and Easywriter are very good, unless you happen to have bought Multiplan and Wordplus-PC, or whatever. Also, although the book is illustrated with screen photos these are very badly taken, and the cover picture, supplied by IBM, is awful.

If you really do have VisiCalc, then perhaps you want *The VisiCalc Book for the IBM Personal Computer*, by Donald Bell. It condenses a mass of instruction into around 340 pages. As far as I can see VisiCalc does not much care what it runs on, it always works in the same way. So while this may be a useful book it is hard to see the addition of the IBM name as much more than a marketing ploy. I have found the VisiCalc manual provides more

information than I actually need, though people who want to push the program to its limits will be glad of the extra help.

IBM's Personal Computer is completely different, and I found it valuable. It is the book to buy before you buy an IBM PC, because it provides all the technical information you need. It shows how the PC fits into IBM's product range and how it fits into the micro market. It provides a full specification of the system with illustrations, plus good descriptions of the systems software and communications protocols. It also methodically evaluates some of the software: VisiCalc, Easywriter and the Peachtree series, plus a few small programs including games.

The resulting volume would be useful to an established data processing department thinking of adding PCs, or to a business user who is about to acquire one. Because it deals with warranties and sales outlets the book's American origins are sometimes a limitation, but otherwise this is a very handy book to have around.

The Executive's Guide to the IBM Personal Computer is clearly no ordinary book. The title is majestic. This ring-bound manual comes in a slip case like a software package with two floppy discs in a holder at the back.

But as I started to flick through it, I had an overwhelming feeling of *déjà vu*. Had I just seen too many IBM PC books? No, I really had read it before. It seems to be page-for-page exactly the same as *Basic for Business*, reviewed here, except that the discs and package inflate the price from £12.70 to £33.95. A

Basic for Business for the IBM PC by Alan J Parker. Published by Reston Publishing, Prentice/Hall International, £12.70. ISBN 0 8359 0355 9.

Hands-On Basic for the IBM Personal Computer by Herbert Peckham. Published by McGraw-Hill, £16.50. ISBN 0 07 049178 X.

IBM Basic for Business and Home by Robert Funkhouser. Published by Reston Publishing, Prentice/Hall International, £12.70. ISBN 0 8359 3018 1.

IBM Data Files: A Basic Tutorial by David Miller. Published by Reston Publishing, Prentice/Hall International, £12.75. ISBN 0 8359 3026 2.

IBM Personal Computer: An introduction to Programming and Applications by Larry Joel and Martin Goldstein. Published by Robert J Brady, Prentice/Hall International, £13.35 or £27.95 including disc. ISBN 0 89303 111 9.

IBM's Personal Computer by DeVoney and Summe. Published by Que Corporation, distributed in the U.K. by The Computer Bookshop, £10.45. ISBN 0 88022 100 3.

Learning IBM Basic for the Personal Computer by David A Lien. Published by Compusoft Publishing, 1050-E Pioneer Way, El Cajon, California CA92020, \$19.95. ISBN 0 932760 13 9.

Some Common Basic Programs: IBM Personal Computer Edition by Poole, Borchers and Burke. Published by Osborne/McGraw-Hill, £12.50. ISBN 0 931988 83 7.

The Executive's Guide to the IBM Personal Computer by Alan J Parker. Published by Reston Publishing, Prentice/Hall International, £33.95. ISBN 0 8359 1809 2.

The VisiCalc Book for the IBM Personal Computer by Donald H. Bell. Published by Reston Publishing, Prentice/Hall International, £13.60. ISBN 0 8359 8395 1.

Useful Basic Programs for the IBM PC by Stanley R Trost. Published by Sybex Inc. £7.95. ISBN 0 89588 111 X.

Using the IBM Personal Computer by T G Lewis. Published by Reston Publishing, Prentice/Hall International, £11.95. ISBN 0 8359 8138 X.

Using Your IBM Personal Computer by Lon Poole. Published by Howard W Sams, Prentice/Hall International, £14.40. ISBN 0 672 22000 8.

Hello Houston, we have a problem

Chris Naylor tells how to enhance cognitive appraisals with a knowledge update

THOSE OF US who are sufficiently grey-haired and decrepit to remember the heady days of space exploration may recall a particular conversation which took place on April 13th, 1970.

"Hey, we've got a problem here!"

"This is Houston. Say again, please."

"Houston, we've had a problem. We've had a main bus B interval."

"Roger. Main B interval. OK. Standby 13. We're looking at it."

"OK. Right now, Houston, the voltage is looking good . . . We had a pretty large bang associated with the caution and warning there. And, if I recall, main B had an amp spike on it once before."

"Roger, Fred."

The exchange took place between Apollo 13 en route to the moon and ground control in Houston and the question that then sprang to the mind of the listener was: What exactly, or even roughly, was it all about? The "problem" was, in fact, that Apollo 13 had just blown up. There is something about the American way of describing things that is singularly impenetrable and, of late, this Houston-ese has been creeping into the computer world at an alarming rate. The reason is simple: nobody has anything interesting to say about computers, but everyone wants to create the opposite impression.

Computers, unlike spacecraft, rarely go to the moon and rarely blow up. The things they do are relatively commonplace, and to state the truth of the situation tends to deprive the speaker of the glamour which he or she may feel to be their due.

The problem is most severe for newcomers to the game. They themselves cannot always see through the speech-opacity of the experienced computer-person and, worse, they have a limited capacity for generating opaque speech, which makes it difficult for them to join in the game. So here, by way of education, is a typically workaday example of how you should, really, explain computers.

We wrote a program. We thought it would work . . .

"We have approached the problem with a real-world orientation and come to the following conclusions. That the problem, as a problem, possessed an implementable structure not limited to the realm of theoretically possible machines but including, also, realisable machines. That, of those realisable machines, at least one such machine had been realised in fact and that a mapping of the problem from the abstract domain into the domain of this realised machine was, in fact, feasible given the right approach. Further, we believed that such a mapping would produce a specific solution which would prove to be both time and space feasible in the new domain. With this in mind we moved at once towards an implementation-achieved type of goal in order to generate a suitable test pattern of theory against a reality-based solution."

. . . but the program was too big.

"At this stage in the process, run-time parameters revealed that the real-world implementation was, initially, alpha-complex to a degree that imposed constraints. By alpha-complex, if we may define a few terms, we mean that a minimal string representation of the problem with no time requirement for implementation was space-infeasible."

We tried to get it to work. . .

"The problem then became one of attempting for a minimax solution in which both the maximum alpha-complexity and the maximum beta-complexity were both simultaneously held to a minimum compatible with execution in the original problem domain. We were motivated in this by a belief that the problem in hand was, at least, semi-tractable."

. . . and it is too slow.

"Moving next to a space-minimal representation with no upper bound to the space requirements revealed a situation in which the implementation was beta-complex, again to an extent that imposed constraints of an unacceptable nature. By beta-complex, we naturally mean that a solution based on a minimal time requirement with an unbounded space requirement lead to a minimal string representation of the second type."

Unfortunately the manual is not clear . . .

"Using paperware look-up we attempted to get a better fix on the specific sub-problem domain by recourse to existing bodies of knowledge whereupon it appeared that the exact sub-problem was one of a class not covered within the general domain of paperware solutions."

. . . which is funny, because we wrote it.

"This produced some cause for internal

consultation and investigation with respect to paperware origination in the hope of pre-empting further situations that might be classified as similar."

We could try a different problem . . .

"Alternatively, we could go for a minimax solution to both the problems of alpha complexity and beta complexity in which the representing string was also current hardware feasible thus allowing a shift in the initial problem domain into the area of that class of problems which have epistemologically adequate solution representations in current hardware terms."

. . . but this one has us beaten.

"Given the foregoing remarks, we are inclined to think that the problem may belong to a class of genuinely hard problems for which no epistemologically adequate solution exists which is both time-minimal and space-minimal due to the problem's alpha-complexity and beta-complexity. Further, should a heuristically adequate representation exist then we doubt that such a representation would genuinely map on to the problem domain in question in a sufficiently thoroughgoing fashion to permit of adequate reliability."

Our invoice will be with you in the morning.

"Related to the foregoing remarks we would note that a paperware solution does exist in relation to the sub-problem of perceived fiscal constraints inherent in a project of this sort and that this solution is both space and time feasible in relation to yourself. And that the sort of, approximate, timescale envisaged is little more than a standard reckoning of twenty-four hours. This particular aspect of the problem may seem semi-hard, but we assure you that it is, in every sense, tractable." 

NEW - ZORBA 2000 SERIES NOW WITH FULL 9" SCREEN



NEW MODEL 4 8 Bit Processor
Full 80x25 9" Screen
800KBytes Disk Storage

NEW MODEL 8 8 Bit Processor
Full 80x25 9" Screen
Massive 1.6 MBytes
Disk Storage

NEW MODEL 16 8 and 16 Bit Processors
Full 80x25 9" Screen
Massive 1.6 MBytes
Disk Storage

All Models Include:-

FREE SOFTWARE

C BASIC ★ WORDSTAR ★ MAILMERGE ★
CALCSTAR

MORE COMPATABILITY

Read & Write the following formats:- IBM PC,
OSBORNE, SUPERBRAIN, XEROX 820,
DEC VT180, KAYCOMP & TELEVIDEO

MORE EXPANDABILITY

Outputs include:- Parallel, Serial & IEEE 488

Details from the Sole UK Importer:-

Sun Computing Services Ltd,
Concorde House, St. Anthony's Way, Feltham,
Middlesex, TW14 0NH.
Tel 01 890 1440
Twx 8954428 SUNCOM G

ZORBA is a registered trademark of MODCOMP INC.

● Circle No. 242